Errata for the

PCI Express Base Specification, Revision 1.0a

28 October 2004

C1 – PWG Approved 5 Jun 2003
C2 – PWG Approved 5 Jun 2003
C3 – PWG Approved 5 Jun 2003
C4 – EWG Approved 10 July 2003
C5 – PWG Approved 5 Jun 2003
C6 – EWG Approved 21 Aug 2003
C7 – PWG Approved 25 Sep 2003
C8 – PWG Approved 16 Oct 2003
C9 – PWG Approved 16 Oct 2003
C10 – PWG Approved 16 Oct 2003
C11 – PWG Approved 16 Oct 2003
C12 – PWG Approved 16 Oct 2003
C13 – SWWG Approved 29 Oct 2003
C14 – EWG Approved 22 Jan 2004
C15 – EWG Approved 22 Jan 2004
C16 – PWG Approved 4 Mar 2004
C17 – PWG Approved 4 Mar 2004
C18 – PWG Approved 23 Feb 2004
C19 – PWG Approved 11 Mar 2004
C19 – SWWG Approved 14 Apr 2004
C20 – PWG Approved 23 Feb 2004
C21 – PWG Approved 22 Apr 2004
C22 – PWG Approved 4 Mar 2004
C23 – PWG Approved 4 Mar 2004
C24 – EWG Approved
C25 – PWG Approved 23 Feb 2004
C26 – PWG Approved 4 Mar 2004
C27 – Withdrawn
C28 – PWG Approved 18 Mar 2004
C29 – PWG Approved 8 Apr 2004
C30 – SWWG Approved 21 Apr 2004
C31 – SWWG Approved 9 May 2004
C32 – PWG Approved 20 May 2004
C33 – SWWG Approved 19 May 2004
C34 – SWWG Approved 9 May 2004
C35 – PWG Approved 20 May 2004
C36 – PWG Approved 24 Jun 2004
C37 – Withdrawn
C38 – Deleted [Duplicate]
C39 – Deleted [Duplicate]
C40 – PWG Approved 24 Jun 2004
C41 – EWG Approved 10 Jun 2004
C42 – PWG Approved 1 Jul 2004
C43 – PWG Approved 24 Jun 2004
C44 – EWG Approved 10 Jun 2004
C45 – EWG Approved 8 Jul 2004
C46 – SWWG Approved 30 Jun 2004
C47 – SWWG Approved 6 July 2004
C48 (E1) – Approved in EWG 9/30/04
C49 (E2) – Approved in EWG 9/30/04
C50 (E3) – Approved in EWG 9/30/04
C51 (E4) – Approved in EWG 10/14/04
C52 (E5) – Approved in EWG 10/14/04
C53 (E6) – Approved in EWG 10/14/04
C54 (E7) – Approved in EWG 10/14/04
C55 (E8) – Approved in EWG 10/14/04
C56 (Cx1) – PWG Approved 14 Oct 2004
C57 (Cx2) - PWG Approved 28 October 2004

REVISION	REVISION HISTORY	DATE
-	First Draft	15 May 2003
	Second Draft	27 May 2003
	Third Draft Approved by Protocol WG with one minor edit 5 Jun 2003	30 May 2003
C1-C5	Fourth Draft/Released Version Combined with material approved by Electrical WG	14 July 2003
C1-C7	Added C6 & C7	7 Oct 2003
C8-15	Modified C2, C4 & C6. Added C8 through C15.	10 Feb 2004
C15-C47	Added C15 through C47	13 Jul 2004
C48-C57 (E1-8 & Cx1-2)	Errata found subsequent to 1.1 Review Draft release	28 Oct 2004

Table of Contents

C1. Max_Payload_Size is DW Based

PWG Approved 5 Jun 2003 -- Release Date: 14 Jul 2003

In Section 2.2.2, "TLPs with Data Payloads – Rules":

- The Transmitter of a TLP with a data payload must not allow the data payload length as given by the TLP's Length[ ] field to exceed the length specified by the value in the Max_Payload_Size field of the Transmitter's Device Control register taken as an integral number of DW (see Section 7.8.4).

...

- The size of the data payload of a Received TLP as given by the TLP's Length[ ] field must not exceed the length specified by the value in the Max_Payload_Size field of the Receiver's Device Control register taken as an integral number of DW (see Section 7.8.4).

C2. Error Reporting and Logging (updated)

PWG Approved 5 Jun 2003 -- Release Date: 14 Jul 2003

Modified and Re-approved by PWG 16 Oct 2003 – Release Date: 22 Jan 2004

This section includes errata regarding error reporting (ERR_* messages) and logging (reporting bits) and the controls associated with these functions.

In section 6.2.7.1, edit the last bullet:

ERR_COR, ERR_NONFATAL and ERR_FATAL are forwarded from the secondary interface to primary interface~~,~~ if the SERR# Enable bit in the ~~Command and~~ Bridge Control register is set. Transmission of forwarded ERR_NONFATAL and ERR_FATAL messages by the primary interface is controlled by the SERR# Enable bit in the Command register.

In section 7.5.1.1, change the bit 8 description:

Table 7-3: Command Register

Bit Location	Register Description	Attributes
...		
8	**SERR Enable** – See Section 7.5.1.7. This bit, ... (see Section 7.8.4). In addition, for Type 1 configuration space header devices, this bit, when set, enables transmission by the primary interface of ERR_NONFATAL and ERR_FATAL error messages forwarded from the secondary interface. This bit does not affect the transmission of forwarded ERR_COR messages. Default value of this field is 0.	RW
...		

Naming inconsistency - Change "SERR Enable" to "SERR# Enable" throughout document

Section 2.3.1, Page 82 line 11:

Note that it is only legal to respond with a configuration retry completion status in response to a

Configuration Request. Sending this Completion Status in response to any other Request type is illegal ~~will result in the generation of an error condition~~ (~~Malformed TLP –~~ see Section ~~6.2~~ 2.3.2).

Section 2.3.2, Page 89 line 20:

Completions with a Configuration Request Retry Status in response to a Request other than a

Configuration Request are illegal. Receivers may optionally report these violations as ~~a~~ Malformed TLPs

• This is a reported error associated with the Receiving Port (see Section 6.2)

In 6.2.4, "Error Logging":

The following PCI Express errors are not function-specific:

...

~~The following are~~These Transaction Layer errors:

Regarding Parity (Poisoned TLP) reporting -

Section 6.2.7.1, second bullet, third sub-bullet:

Change "Parity Error Response" to "Parity Error Response Enable"

Section 6.2.7.1, second bullet, last sub-bullet:

Change "Bridge Control register" to "Command register"

Section 7.5.1.1, bit 6:

Change "Parity Error Enable" to "Parity Error Response"

Section 7.5.1.2, bit 8 (Master Data Parity Error) - change:

"if ~~its Parity Error Enable bit is set~~ the Parity Error Response bit in the Command register is 1b"

... "If the Parity Error ~~Enable bit is cleared~~ Response bit is 0b".

Section 7.5.1.2, bit 15 (Detected Parity Error):

Change "Parity Error Enable bit" to "Parity Error Response bit in the Command register"

Section 7.5.1.7, first paragraph:

The error control/status register bits in the Command and Status registers (see Section 7.5.1.1 and Section 7.5.1.2 respectively) and the Bridge Control and Secondary Status registers of Type 1 Configuration Space header devices (see Section 7.5.3.5 and Section 7.5.3.3 respectively) control PCI compatible error reporting for both PCI and PCI Express devices.

Section 7.5.3.3, bit 8 (Master Data Parity Error):

"if the Parity Error Response ~~bit is set~~ Enable bit in the Bridge Control register is 1b".

... "If the Parity Error Response ~~bit is cleared~~ Enable bit is 0b".

Section 7.5.3.3, bit 15 (Detected Parity Error):

"Parity Error Response ~~bit~~ Enable bit in the Bridge Control register"

Section 7.5.3.3, bit 14 (Received System Error):

"when the Secondary Side for a Type 1 configuration space header device receives an ~~a device sends a~~ ERR_FATAL or ERR_NONFATAL Message"

C3. Root Port Arbitration [modified]

PWG Approved 5 Jun 2003 -- Release Date: 14 Jul 2003
Modified version: PWG Approved 24 Jun 2004 -- Release Date: 12 Jul 2004

p 282 - 6.3.3.3. Port Arbitration – Arbitration Within VC – Edit text as shown:

For switches, Port Arbitration ~~within a VC~~ refers to the arbitration at an Egress Port between ~~the~~ traffic coming from other Ingress Ports that is mapped ~~on~~to the same VC ~~but is coming from different Ingress Ports~~. For Root Ports, Port Arbitration refers to the arbitration at a Root Egress Port between peer-to-peer traffic coming from other Root Ingress Ports that is mapped to the same VC. For RCRBs, Port Arbitration refers to the arbitration at the RCRB (e.g., for host memory) between traffic coming from Root Ports that is mapped to the same VC.

Inherent prioritization scheme ...

p 283 - 6.3.4.1. Rules for Software Configuration – Edit text as shown:

System software must obey the following rules to configure PCI Express fabric for isochronous traffic:

...

- For each Switch Egress Port and RCRB that supports isochronous data flows, the associated Port Arbitration must support time-based arbitration for the configured VC. This requirement also applies for Root Complexes that support isochronous data flows peer-to-peer between Root Ports.
- Software should not intermix isochronous traffic with non-isochronous traffic on a given VC.

p 284 - 6.3.4.4. Rules for Switch Components - Edit text as shown:

6.3.4.4. Rules for Switch and Root Complex Components

A Switch component providing isochronous services must obey the following rules. The same rules apply to Root Complexes that support isochronous data flows peer-to-peer between Root Ports, abbreviated in this section as "P2P-RC".

- An isochronous-configured Switch or P2P-RC Port should not apply flow control induced backpressure to uniformly injected isochronous requests under normal operating conditions.
- An isochronous-configured Switch or P2P-RC Port must observe the maximum isochronous transaction latency.
- A Switch or P2P-RC component must support time-based Port Arbitration mechanism for each Port that supports one or more VCs capable of supporting isochronous traffic. Note that the time-based Port Arbitration only applies to request transactions but not to completion transactions.

p 380-381 - Table 7-35: Port VC Capability Register 1 – Edit text as shown:

Table 7-35: Port VC Capability Register 1

Bit Location	Register Description	Attributes
9:8	**Reference Clock** – Indicates the reference clock for Virtual Channels that support time-based WRR Port Arbitration. This field is valid for RCRBs, ~~and~~ for Switch Ports, and for Root Ports that support peer to peer traffic; it is not valid for ~~Root Ports and~~ Endpoint devices and must be set to 0. Defined encodings are: 00b 100 ns reference clock 01b – 11b Reserved	RO
11:10	**Port Arbitration Table Entry Size** – Indicates the size (in bits) of Port Arbitration table entry in the device. This field is valid only for RCRBs, ~~and any~~ for Switch Ports, and for Root Ports that support peer to peer traffic. It is not valid and must be set to 0 for Endpoint devices ~~and Root Ports~~. ...	RO

p 384-385 - Table 7-39: VC Resource Capability Register – Edit text as shown:

Table 7-39: VC Resource Capability Register

Bit Location	Register Description	Attributes
7:0	**Port Arbitration Capability** – Indicates types of Port Arbitration supported by the VC resource. This field is valid for all Switch Ports, Root Ports that support peer to peer traffic, and RCRBs, but not for PCI Express Endpoint devices or Root Ports that do not support peer to peer traffic. ...	RO
…	…	…
22:16	**Maximum Time Slots** – Indicates ... This field is valid for all Switch Ports, Root Ports that support peer to peer traffic, and RCRBs, but not for PCI Express Endpoint devices or Root Ports that do not support peer to peer traffic. ...	HwInit

Bit Location	Register Description	Attributes
31:24	**Port Arbitration Table Offset** – Indicates the location of the Port Arbitration Table associated with the VC resource. This field is valid for all Switch Ports, Root Ports that support peer to peer traffic, and RCRBs, but not for PCI Express Endpoint devices or Root Ports that do not support peer to peer traffic.	RO

p 386 - Table 7-40: VC Resource Control Register – Edit text as shown:

Table 7-40: VC Resource Control Register

Bit Location	Register Description	Attributes
16	**Load Port Arbitration Table** – This bit, when set, updates the Port Arbitration logic from the Port Arbitration Table for the VC resource. This field is valid for all Switch Ports, Root Ports that support peer to peer traffic, and RCRBs, but not for PCI Express Endpoint devices or Root Ports that do not support peer to peer traffic.	RW
19:17	**Port Arbitration Select** – This field configures the VC resource to provide a particular Port Arbitration service. This field is valid for RCRBs, Root Ports that support peer to peer traffic, and Switch Ports, but not for PCI Express Endpoint devices or Root Ports that do not support peer to peer traffic. ...	RW

p 388 - Table 7-41: VC Resource Status Register – Edit text as shown:

Table 7-41: VC Resource Status Register

Bit Location	Register Description	Attributes
0	**Port Arbitration Table Status** – This bit indicates the coherency status of the Port Arbitration Table associated with the VC resource. This field is valid for RCRBs, Root Ports that support peer to peer traffic, and Switch Ports, but not for PCI Express Endpoint devices or Root Ports that do not support peer to peer traffic. In addition, this field is valid only when the Port Arbitration Table is used by the selected Port Arbitration for the VC resource. ...	RO

p 395 - 7.11.10. Port Arbitration Table – Edit text as follows:

The Port Arbitration Table register is a read-write register array that is used to store the WRR or time-based WRR arbitration table for Port Arbitration for the VC resource. This register array is valid for all Switch Ports, Root Ports that support peer to peer traffic, and RCRBs, but not for Endpoint devices ~~or Root Ports~~.
...

p 410 - A.8.4. Root Complex – Add a new bullet as follows:

…

- Configuration of the Root Port as an Egress Port:
  - If supported, configuration of the Root Port's VC Arbitration Table with large weights assigned to the associated VC.
  - If the Root Complex supports peer-to-peer traffic between Root Ports, configuration of the Root Port's Port Arbitration Table number of entries is set according to the assigned isochronous bandwidth for all Ingress Ports.

C4. Physical Layer (updated)

EWG Approved 10 July 2003 -- Release Date: 14 Jul 2003

Modified and Re-approved by EWG 22 Jan 2004 – Release Date: 22 Jan 2004

Section 4.2.6.2.4*. This Note is to be inserted at the end of this section:*

Implementation Note: Multiple Data Rates Supported Within One Port

There is optional behavior allowed in Polling. Speed. In the event that only some lanes are sending and receiving data rates identifiers with higher data rates <insert cross-reference to tables 4-2 an 4-3>, training may continue with a single LTSSM as described above or optionally, additional LTSSMs can be initiated with lanes divided between the various LTSSMs based on the highest supported common data rate.

In Section 4.2.6.3.1.1: (5th bullet), edit as shown:

~~The Transmitter sends TS1 ordered sets with selected Link numbers and sets Lane numbers to PAD (K23.7) on Downstream Lanes that are in Configuration.~~ The Transmitter sends TS1 ordered sets with selected Link numbers and sets Lane numbers to PAD (K23.7) on all Downstream Lanes that detected a receiver during Detect.

Section 4.2.6.3.1.1 *The Note is to be inserted at the end of this section; (after 7th bullet):*

Implementation Note: Crosslink Initialization

In the case where the Downstream Lanes are connected to both Downstream Lanes (crosslink) and Upstream Lanes, the port with the Downstream Lanes may continue with a single LTSSM as described in this section or optionally, split into multiple LTSSMs.

Section 4.2.6.3.3 Configuration.Lanenum.Accept: *edit as shown:*

4.2.6.3.3.1 Downstream Lanes

- If two consecutive …
  - Note: Reversed Lane numbers …
  - Note: It is recommended that …
- If a configured Link can be formed with any subset of the Lanes that receive two consecutive TS1 ordered sets with the same transmitted non-PAD Link numbers and any non-Pad Lane numbers, TS1

ordered sets are transmitted with new Lane numbers assigned and the next state is Configuration.Lanenum.Wait.

- Note: The newly assigned transmitted Lane numbers must range from 0 to m-1, be assigned sequentially only to some continuous grouping of the Lanes that are receiving non-PAD Lane numbers (i.e., Lanes which are not receiving any TS1 ordered sets always disrupt a continuous grouping and must not be included in this grouping), must include either Lane 0 or n-1 (largest received Lane number), and m-1 must be equal to or smaller than the largest received Lane number (n-1). Any left over Lanes must transmit TS1 ordered sets with the Link and Lane number set to PAD (K23.7).

…

4.2.6.3.3.2. Upstream Lanes

- If two consecutive TS2 ordered sets …
- If a configured Link can be formed with any subset of the Lanes that receive two consecutive TS1 ordered sets with the same transmitted non-PAD Link numbers and any non-PAD Lane numbers, TS1 ordered sets are transmitted with new Lane numbers assigned and the next state is Configuration.Lanenum.Wait.

  - Note: The newly assigned transmitted Lane numbers must range from 0 to m-1, be assigned sequentially only to some continuous grouping of Lanes that are receiving non-PAD Lane numbers (i.e., Lanes which are not receiving any TS1 ordered sets always disrupt a continuous grouping and must not be included in this grouping), must include either Lane 0 or n-1 (largest received Lane number), and m-1 must be equal to or smaller than the largest received Lane number (n-1). Any left over Lanes must transmit TS1 ordered sets with the Link and Lane number set to PAD (K23.7).

…

Configuration.Lanenum.Wait

4.2.6.3.4.1. Downstream Lanes:

- The next state is Configuration.Lanenum.Accept if any of the Lanes that detected a receiver during Detect receive two consecutive TS1 which have a Lane number different from when the Lane first entered Configuration.Lanenum.Wait, and not all the Lanes' Link numbers are set to Pad (K23.7).

Note: The Upstream Lanes may delay up to 1 msec before transitioning to Configuration.Lanenum.Accept.

The reason for delaying up to 1msec before transitioning is to prevent received errors or skew between Lanes affecting the final configured Link width.

The condition of requiring reception of any Lane number different from when the Lane(s) first entered Configuration.Lanenum.Wait is necessary in order to allow the two ports to settle on an agreed upon Link width. The exact meaning of the statement "any of the Lanes receive two consecutive TS1s, which have a Lane number different from when the Lane first entered Configuration.Lanenum.Wait" requires that a Lane number must have changed from when the Lanes most recently entered Configuration.Lanenum.Wait before a transition to Configuration.Lanenum.Accept can occur.

- The next state is Detect after a 2 ms timeout or if all Lanes receive two consecutive TS1 ordered sets with Link and Lane numbers set to Pad (K23.7).

In 4.2.6.6.1.1. *Rx_L0s.Entry 10 (page 190.):*

- Next state is Rx_L0s.Idle after a $T_{TX\text{-}IDLE\text{-}MIN}$ (<ref>) timeout
  - Note: This is the minimum time the Transmitter must be in an Electrical Idle condition.

Section 4.2.6.9 *- Change the first bullet to say the following "All Lanes transmit 16 to 32 TS1 ordered sets with the Disable Link bit (bit 1) asserted and then transition to Electrical Idle.". The change is outlined in bold and addresses the issue that the ability to send 16 TS1s with precision is not necessary and adds complexity.*

- All Lanes transmit 16 to 32 TS1 ordered sets with the Disable Link bit (bit 1) asserted and then transition to Electrical Idle.

Between section 4.2.6.9 and 4.2.6.9.1. *A Heading was dropped between section 4.2.6.9 and 4.2.6.9.1. (Page 196) Below shows what the spec should have looked like. Also note that this would have changed the 4.2.xsections to 4.2.x where x is 10 or higher.*

4.2.6.10 Loopback

The Loopback sub-state machine is shown in <ref>

4.2.6.10.1 Loopback.Entry

Insert red text shown below at the end of section 4.3.2.4:

For the case described above of Beacon pulses with a width greater than 500 ns, the minimum beacon amplitude is –6 dB down from the minimum differential peak to peak output voltage ($V_{TX\text{-}DIFFp\text{-}p}$). The maximum beacon amplitude for this case is –6 dB down from the maximum peak to peak output voltage ($V_{TX\text{-}DIFFp\text{-}p}$).

C5. Miscellaneous Errata

PWG Approved 5 Jun 2003 -- Release Date: 14 Jul 2003

In Section 1.3.3, last bullet: "may not" -> "must not"

In Section 6.7.3.5, last paragraph: "as defined" -> "such as defined"

In 2.2.6.2:

- Prior to the initial Configuration Write to a device, the device is not permitted to initiate Non-Posted Requests (A valid Requester ID is required to properly route the resulting completions).

In terms, edit as shown:

reserved — The contents, states, or information are not defined at this time. Using any reserved area (for example, packet header bit-fields, configuration register bits) is not permitted. Reserved register fields must be read only and must return 0 when read. Reserved encodings for register and packet fields must not be used. Any implementation dependence on a reserved field value or encoding will result in an implementation that is not PCI Express-compliant. The functionality of such an implementation cannot be guaranteed in this or any future revision of this specification.

In Section 7.4, Table 7-2, edit as shown:

Table 0-1: Register (and Register Bit-Field) Types

Register Attribute	Description
...	
RsvdP	Reserved and Preserved: Reserved for future RW implementations~~;~~. Registers are read-only and must return 0 when read. S~~s~~oftware must preserve value read for writes to bits.
RsvdZ	Reserved and Zero: Reserved for future RW1C implementations. Registers are read-only and must return 0 when read. S~~; s~~oftware must use 0 for writes to bits.

In Section 2.2.5, "First/Last DW Byte Enable Rules":

...

- Non-contiguous Byte Enables are permitted in both Byte Enables fields for QW aligned Memory Requests with length of 2 DW (1 QW).
- All non-QW aligned Memory Requests with length of 2 DW (1 QW) and Memory Requests with length of 3 DW or more must enable only bytes that are contiguous with the data between the first and last DW of the Request.

In Section 5.4.1, bullets below the implementation note, change text as shown:

- Functions in a non-D0 state (D1 and deeper) are ignored in determining the ASPM policy

- If any of the functions in the D0 state has its ASPM disabled (ASPM Control field = 00b), or if at least one of the functions in the D0 state is enabled for L0s only (ASPM Control field = 01b) and at least one other function in the D0 state is enabled for L1 only (ASPM Control field = 10b), then ASPM is disabled for the entire component

- Else, if at least one of the functions in the D0 state is enabled for L0s only (ASPM Control field = 01b), then ASPM is enabled for L0s only

- Else, if at least one of the functions in the D0 state is enabled for L1 only (ASPM Control field = 10b), then ASPM is enabled for L1 only

- Else, ASPM is enabled for both L0s and L1 states

In Section 5.1, delete as shown:

Once in the lower power state, transitions to the fully operative L0 state are triggered by traffic appearing on either side of the Link. ~~Endpoints initiate entry into a low power Link state. This feature~~ ASPM may be disabled by software. Refer to Section 5.4.1 for more information on ASPM.

C6. Loopback (updated)

EWG Approved 21 Aug 2003 -- Release Date: 7 Oct 2003

Modified and Re-approved by EWG 22 Jan 2004 – Release Date: 22 Jan 2004

Replace Section 4.2.6.9.2 with the following:

4.2.6.9.2. Loopback.Active

The Loopback Master must send valid 8b/10b data. The next state of the Loopback Master is Loopback.Exit if directed.

A Loopback Slave is required to retransmit the received 10-bit information as received, with the polarity inversion determined in Polling applied, while continuing to perform clock tolerance compensation:

SKPs must be added or deleted on a per Lane basis as outlined in Section 4.2.7 with the exception that SKPs do not have to be simultaneously added or removed across Lanes of a configured Link.

If a SKP ordered set retransmission requires adding a SKP Symbol to accommodate timing tolerance correction, the SKP Symbol is inserted in the retransmitted Symbol stream anywhere adjacent to a SKP Symbol in the SKP ordered set following the COM Symbol. The inserted SKP Symbol must be of the same disparity as the received SKPs Symbol(s) in the SKP ordered set.

If a SKP ordered set retransmission requires dropping a SKP Symbol to accommodate timing tolerance correction, the SKP Symbol is simply not retransmitted.

Note: No modifications of the received 10-bit data (except for polarity inversion determined in Polling) is allowed by the Loopback Slave even if
it is determined to be an invalid 10-bit code (i.e., no legal translation to a control or data value possible).

Next state of the Loopback Slave is Loopback.Exit when an Electrical Idle ordered set is received or Electrical Idle is detected on any Lane.

Note: A Loopback Slave must be able to detect an Electrical Idle condition on any Lane with 1 ms of the Electrical Idle ordered set being received by the Loopback Slave.

Note: During the time after an Electrical Idle ordered set is received and before Electrical Idle is actually detected by the Loopback Slave, the Loopback Slave may receive and transmit undefined 10-bit data.

The TTX-IDLE-SET-TO-IDLE parameter does not apply in this case since the Loopback Slave may not even detect Electrical Idle until as much as 1 ms after the Electrical Idle ordered set.

The next state of the Loopback Master is Loopback.Exit if directed.

C7. ASPM & PCI-PM L1

PWG Approved 25 Sep 2003 -- Release Date: 7 Oct 2003

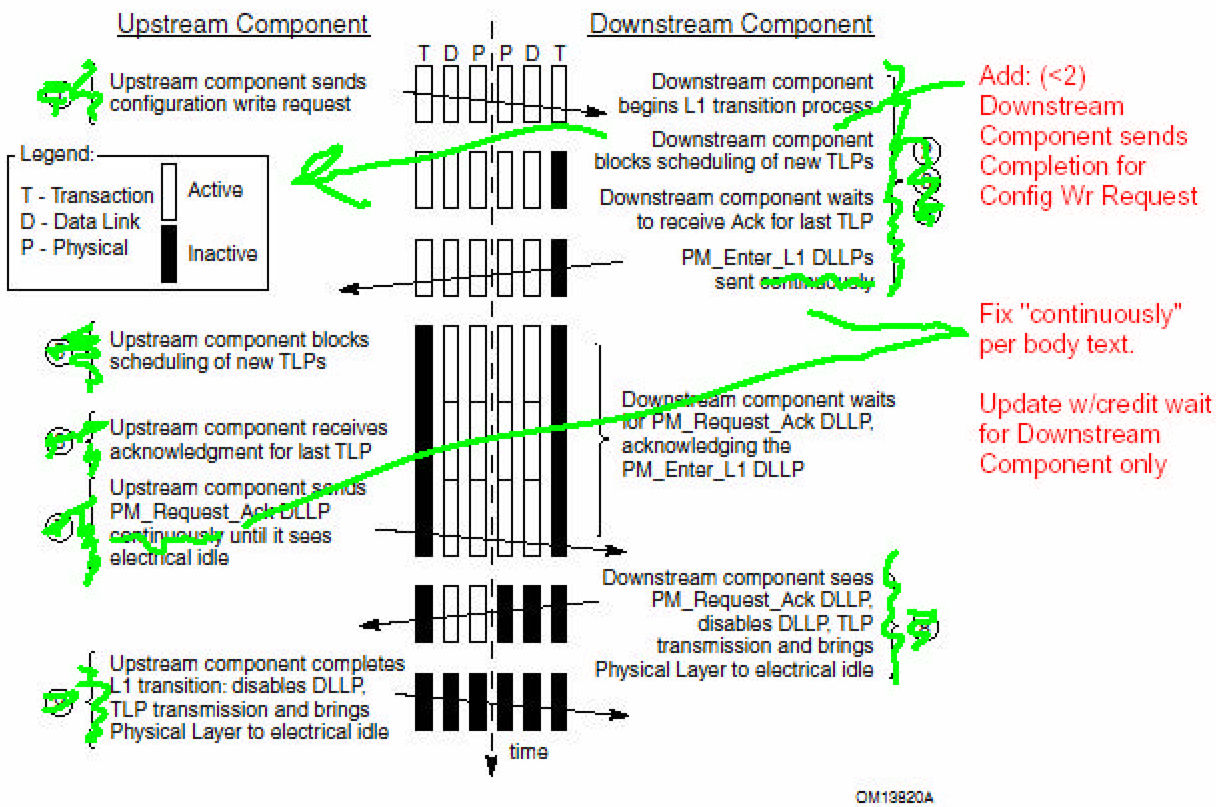


Figure 5-2: Entry into L1 Link State

In 5.3.2.1. Entry into the L1 State:

...

PM Software Request:

1. PM Software ~~(Upstream component)~~ sends a TLP configuration ~~request packet~~ write to the Downstream function's PMCSR to change the Downstream function's D-state (from D0 to D1 for example).

Downstream Component Link State Transition Initiation Process:

2. The Downstream component schedules the completion response corresponding to the configuration write to its PMCSR PowerState field and accounts for the completion credits required.

3. The Downstream component must then wait until it accumulates at least the minimum number of credits required to send the largest possible packet for any FC type (if it does not already have such credits). All ~~new~~ Transaction Layer TLP scheduling is then suspended.

~~4.~~ The Downstream component then waits until it receives a Link Layer acknowledgement for the PMCSR write completion, and any other TLPs it had previously sent. The component ~~may~~ must retransmit a TLP out of its Data Link Layer Retry buffer if required to do so by Data Link Layer rules. ~~The Downstream component must also wait until it accumulates at least the minimum number of credits required to send the largest possible packet for any FC type. Note that this is required so that the component can immediately issue a TLP after it exits the L1 state.~~

5. Once all of the Downstream component's TLPs have been acknowledged, the Downstream component starts to transmit~~s a~~ PM_Enter_L1 DLLP~~s onto its upstream-directed (Transmit) Port~~. The Downstream component sends this DLLP ~~continuously~~repeatedly with no more than 4 symbol times of idle between subsequent transmissions of the PM_Enter_L1 DLLP. The transmission of other DLLPs and SKP ordered sets is permitted at any time between PM_Enter_L1 transmissions, and do not contribute to the 4 symbol time idle limit.

   The Downstream component continues to transmit the PM_Enter_L1 DLLP as described above until it receives a response from the Upstream component[2] (PM_Request_Ack). ~~While waiting for all of its TLPs to be acknowledged, the Downstream component must not initiate any new TLPs.~~

   The Downstream component must~~ still, however,~~ continue to accept TLPs and DLLPs from the Upstream component, and it must also continue to respond with DLLPs, including FC update DLLPs and Ack/Nak DLLPs, as required.~~needed per Link Layer protocol. (Refer to Chapter 4 for more details on the Physical Layer behavior).~~ Any TLPs that are blocked from transmission (including responses to TLP(s) received) must be stored for later transmission, and must cause the Downstream Component to initiate L1 exit as soon as possible following L1 entry.

Upstream Component Link State Transition Process:

6. Upon receiving the PM_Enter_L1 DLLP the Upstream component blocks the scheduling of ~~any future~~ all ~~TLPs~~TLP transmissions.

7. The Upstream component then must wait until it receives a Link Layer acknowledgement for the last TLP it had previously sent. The Upstream component ~~may~~ must retransmit a TLP from its Link Layer retry buffer if required to do so by the Link Layer rules. ~~The Upstream component must also wait until it accumulates at least the minimum number of credits required to send the largest possible packet for any FC type. Note that this is required so that the component can immediately issue a TLP after it exits the L1 state.~~

---

[2] If at this point the Downstream component needs to initiate a transfer on the Link, it must first complete the transition to L1 regardless. Once in L1 it is then permitted to initiate an exit L1 to handle the transfer. ~~This corner case represents an event requiring a PME message occurring during the component's transition to L1.~~

8. Once all of the Upstream component's TLPs have been acknowledged the Upstream component ~~sends~~ must send ~~a~~ PM_Request_Ack DLLPs downstream, regardless of any outstanding Requests. The Upstream component sends this DLLP ~~continuously~~repeatedly with no more than 4 symbol times of idle between subsequent transmissions of the PM_Request_Ack DLLP. The transmission of SKP ordered sets is permitted at any time between PM_Request_Ack transmissions, and does not contribute to the 4 symbol time idle limit.

   The Upstream component continues to transmit the PM_Request_Ack DLLP as described above until it observes its receive Lanes enter into the Electrical Idle state. See Chapter 4 for more details on the Physical Layer behavior.[3]

Completing the L1 Link State Transition:

9. Once the Downstream component has captured the PM_Request_Ack DLLP on its Receive Lanes (signaling that the Upstream component acknowledged the transition to L1 request), it then disables DLLP~~, TLP~~ transmission and brings the upstream directed physical Link into the Electrical Idle state.

10. When the Upstream component observes its Receive Lanes enter the Electrical Idle state, it then stops sending PM_Request_Ack DLLPs, disables DLLP~~, TLP~~ transmission and brings its Transmit Lanes to Electrical Idle completing the transition of the Link to L1.

    When two components' interconnecting Link is in L1 as a result of the Downstream component being programmed to a non-D0 state, both components suspend the operation of their Flow Control Update, ~~DLLP ACK/NAK Latency, and~~ TLP Completion Timeout and, if implemented, Update FCP Timer (see Section 2.6.1.2) counter mechanisms. Refer to the Chapter 4 for more detail on the Physical Layer behavior.

Components on either end of a Link in L1 may optionally disable their internal PLLs in order to conserve more energy. Note, however, that platform supplied main power and reference clocks must continue to be supplied to components on both ends of an L1 Link.

In 5.3.2.2. Exit from L1 State:

L1 exit can be initiated by the component on either end of a PCI Express Link. ~~A Downstream component would initiate an L1 exit transition in order to bring the Link to L0 such that it may then inject a PME Message.~~

~~The Upstream component initiates L1 exit to re-establish normal TLP and DLLP communications on the Link.~~ Upon exit from L1, it is recommended that the Downstream Component send flow control update DLLPs for all enabled VCs and FC types starting within 1us of L1 exit.

~~In either case t~~The physical mechanism for transitioning a Link from L1 to L0 is ~~the same and is~~ described in detail ~~with~~in Chapter 4.

L1 exit must be initiated by a component if that component needs to transmit a TLP on the Link. An upstream component must initiate L1 exit on a downstream port even if it does not have the flow control credits needed to transmit the TLP that it needs to transmit. Following L1 exit the upstream component must wait to receive the needed credit from the downstream component. Figure 5-3 outlines an example sequence that would trigger an Upstream component to initiate transition of the Link to the L0 state.

<figure and example not changed>

[3] ~~If, at this point, the Upstream component for any reason needs to initiate a transfer on the Link, it must first complete the transition to L1 regardless. Once in L1, the component is permitted to initiate an exit from L1 to handle the transfer.~~

...

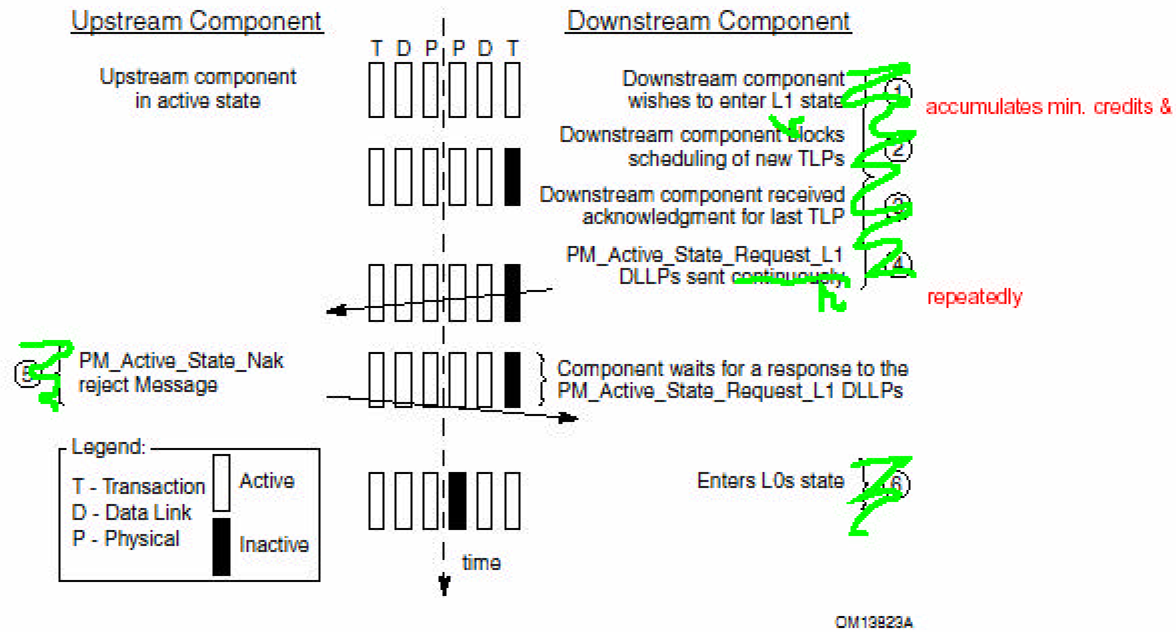


Figure 5-6: L1 Transition Sequence Ending with a Rejection (L0s Enabled)

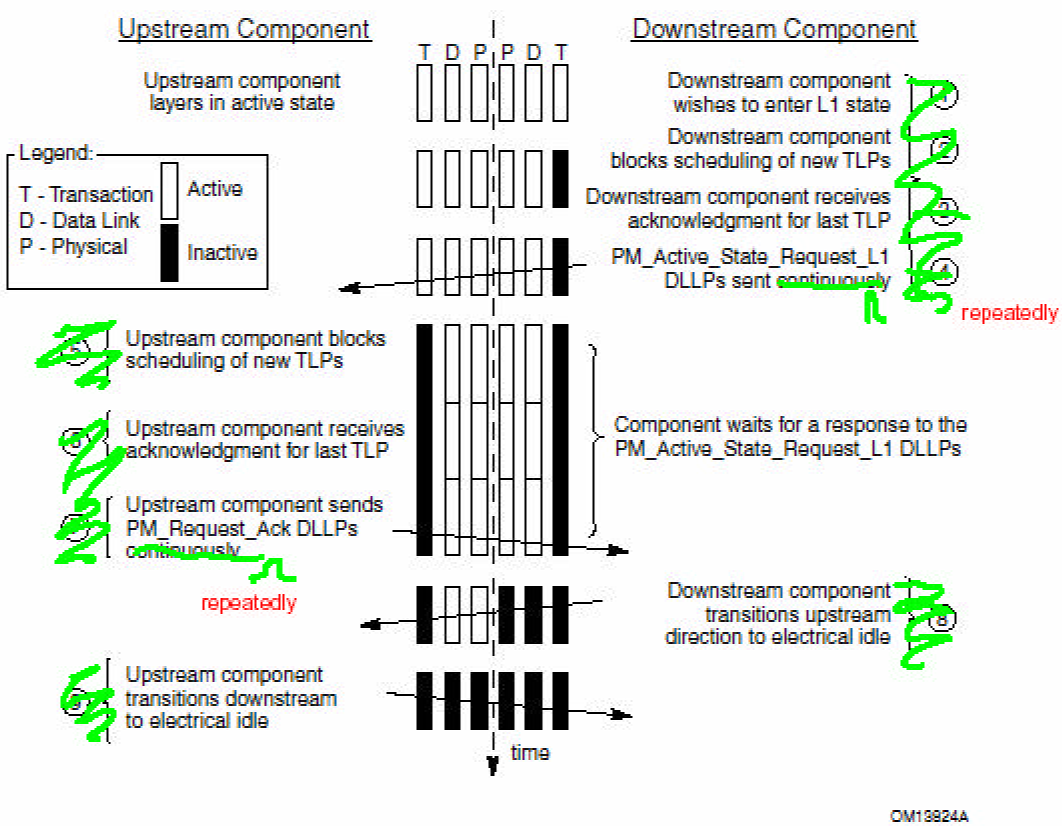


Figure 5-7: L1 Successful Transition Sequence

In 5.4.1.2.1. Entry to L1 State:

…

ASPM L1 Negotiation Rules (see Figure 5-6 and Figure 5-7)

- The Downstream component must not initiate ASPM L1 entry until it accumulates at least the minimum number of credits required to send the largest possible packet for any FC type.
- Upon deciding to enter a low power Link state, the Downstream component must block ~~scheduling~~ movement of all~~any~~ TLPs from the Transaction Layer to the Data Link Layer for transmission (including completion packets).
  - If any ~~new~~ TLPs become available from the Transaction Layer for transmission during the L1 negotiation process, the transition to L1 must first be completed and then the Downstream component must initiate a return to L0.
- The Downstream component must wait until it receives a Link Layer acknowledgement for the last TLP it had previously sent (i.e., the retry buffer is empty). The component ~~may~~ must retransmit a TLP out of its Data Link Layer Retry buffer if required by the Data Link Layer rules.
- ~~The Downstream component must also wait until it accumulates at least the minimum number of credits required to send the largest possible packet for any FC type. Note that this is required so that the component can immediately issue a TLP after it exits the L1 state.~~
- The Downstream component then initiates ASPM negotiation by sending a PM_Active_State_Request_L1 DLLP onto its Transmit Lanes. The Downstream component sends this DLLP ~~continuously~~repeatedly with no more than 4 symbol times of idle between subsequent transmissions of the PM_Active_State_Request_L1 DLLP. The transmission of other DLLPs and SKP ordered sets is permitted at any time between PM_Active_State_Request_L1 transmissions, and do not contribute to the 4 symbol time idle limit.
- The Downstream component continues to transmit the PM_Active_State_Request_L1 DLLP as described above until it receives a response from the Upstream device (see below). The Downstream component remains in this loop waiting for a response from the Upstream Component~~Agent~~.
  - During this waiting period, the Downstream component must not initiate any Transaction Layer transfers. It must still accept TLPs and DLLPs from the Upstream component, storing for later transmission any TLP responses required. It ~~also~~ continues to respond~~s~~ with DLLPs, including FC update DLLPs, as needed by the Link Layer protocol. ~~All received TLPs must be acknowledged before entering the L1 state (i.e., DLLP ACK/NAK Latency timers are reset before entering the L1 state)~~.
  - If the Downstream component for any reason needs to ~~initiate a transfer~~ transmit a TLP on the Link, it must first complete the transition to the low power Link state. Once in a lower power Link state, the Downstream component ~~is~~ must then ~~permitted to~~ initiate exit of the low power Link state to handle the transfer.
- The Upstream component must immediately respond to the request with either an acceptance or a rejection of the request.
  - If the Upstream component is not able to accept the request, it must immediately reject the request.

Rules in case of rejection:

- In the case of a rejection, the Upstream component must schedule, as soon as possible, a rejection ~~(Nak)~~ by sending the PM_Active_State_Nak Message to the Downstream Component~~requesting agent~~. Once the PM_Active_State_Nak Message is sent, the Upstream component is permitted to initiate any TLP or DLLP transfers.
- If the request was rejected, the Downstream component must immediately transition its Transmit Lanes into the L0s state, provided L0s is enabled and that conditions for L0s entry are met.
- Prior to transmitting a PM_Active_State_Request_L1 DLLP associated with a subsequent ASPM L1 negotiation sequence, the Downstream component must either enter and exit L0s on its Transmitter, or it must wait at least 10µs from the last transmission of the PM_Active_State_Request_L1 DLLP associated with the preceding ASPM L1 negotiation. This 10µs timer must count only time spent in the LTSSM L0 and L0s states. The timer must hold in the LTSSM Recovery state. If the Link goes down and comes back up, the timer is ignored and the component is permitted to issue new ASPM L1 request after the link has come back up.

IMPLEMENTATION NOTE

ASPM L1 Accept/Reject Considerations for Upstream Component

When the Upstream component has responded to the Downstream component's ASPM L1 request with a PM_Request_Ack DLLP to accept the L1 entry request, the ASPM L1 negotiation protocol clearly and unambiguously ends with the Link entering L1. However, if the Upstream component responds with a PM_Active_State_Nak Message to reject the L1 entry request, the termination of the ASPM L1 negotiation protocol is less clear. But, both components need to be designed to unambiguously terminate the protocol exchange. If this is not done, there is the risk that the two components will get out of sync with each other, and the results may be undefined. For example, consider the following case:

* The Downstream component requests ASPM L1 entry by transmitting a sequence of PM_Active_State_Request_L1 DLLPs

* Due to a temporary condition, the Upstream component responds with a PM_Active_State_Nak Message to reject the L1 request

* The Downstream component continues to transmit the PM_Active_State_Request_L1 DLLPs for some time before it is able to respond to the PM_Active_State_Nak Message

* Meanwhile, the temporary condition that previously caused the Upstream component to reject the L1 request is resolved, and the Upstream component erroneously sees the continuing PM_Active_State_Request_L1 DLLPs as a new request to enter L1, and responds by transmitting PM_Request_Ack DLLPs downstream

At this point, the result is undefined, because the Downstream component views the L1 request as rejected and finishing, but the Upstream component views the situation as a second L1 request being accepted.

To avoid this situation, the Downstream component needs to provide a mechanism to distinguish between one ASPM L1 request and another. The Downstream component does this by entering L0s or by waiting a minimum of 10µs from the transmission of the last PM_Active_State_Request_L1

DLLP associated with the first ASPM L1 request before starting transmission of the PM_Active_State_Request_L1 DLLPs associated with the second request (as described above).

If the Upstream component is capable of exhibiting the behavior described above, then it is necessary for the Upstream component to recognize the end of an L1 request sequence by detecting a transition to L0s on its receiver or a break in the reception of PM_Active_State_Request_L1 DLLPs of 9.5µs measured while in L0/L0s or more as a separation between ASPM L1 requests by the Downstream component.

If there is a possibility of ambiguity, the Upstream component should reject the L1 request to avoid potentially creating the ambiguous situation outlined above.

Rules in case of acceptance:

- If the Upstream Component~~agent~~ is ready to accept the request, it must block scheduling of any ~~new~~ TLPs from the Transaction Layer.
- The Upstream component then must wait until it receives a Data Link Layer acknowledgement for the last TLP it had previously sent. The Upstream component ~~may~~ must retransmit a TLP if required by the Data Link Layer rules.
- ~~The Upstream component must also wait until it accumulates at least the minimum number of credits required to send the largest possible packet for any FC type. Note that this is required so that the component can immediately issue a TLP after it exits the L1 state.~~
- Once all TLPs have been acknowledged ~~and enough FC credits accumulated~~, the Upstream component sends a PM_Request_Ack DLLP downstream. The Upstream component sends this DLLP ~~continuously~~repeatedly with no more than 4 symbol times of idle between subsequent transmissions of the PM_Request_Ack DLLP. The transmission of SKP ordered sets is permitted at any time between PM_Request_Ack transmissions, and do not contribute to the 4 symbol time idle limit.
- The Upstream component continues to transmit the PM_Request_Ack DLLP as described above until it observes its Receive Lanes enter into the Electrical Idle state. See Chapter 4 for more details on the Physical Layer behavior.
- If the Upstream component needs, for any reason, to ~~initiate a transfer~~ transmit a TLP on the Link after it sends a PM_Request_Ack DLLP, it must first complete the transition to the low power state~~. It is~~, and then ~~permitted to~~ initiate exit from the low power state to handle the transfer once the Link is back to L0.
    - The Upstream component must initiate exit from L1 in this case even if it does not have the required flow control credit to transmit the TLP(s).

...

In 5.4.1.1.2. Exit from L1 State:

Components on either end of a PCI Express Link may initiate an exit from the L1 Link state. ~~Unlike the L1 entry protocol, where both ends negotiate for the resultant Link state, L1 exit does not require any negotiation.~~

Upon exit from L1, it is recommended that the Downstream Component send flow control update DLLPs for all enabled VCs and FC types starting within 1us of L1 exit.

Downstream Component Initiated Exit

An Endpoint or Switch Upstream Port ~~is permitted to~~ must initiate an exit from L1 on its Transmit Lanes if it needs to communicate through the Link. The component initiates a transition to the L0 state as described in Chapter 4. The Upstream component must respond by initiating a similar transition of its Transmit Lanes.

...

Upstream Component Initiated Exit

A Root Complex, or a Switch ~~is permitted to~~ must initiate an exit from L1 on any of its Root Ports, or Downstream Port Links if it needs to communicate through that Link. The Switch or Root Complex must be capable of initiating L1 exit even if it does not have the flow control credits needed to transmit a given TLP. The component initiates a transition to the L0 state as described in Chapter 4. The Downstream component must respond by initiating a similar transition on its Transmit Lanes.

...

C8. Enabling ASPM L1 [typo correction 25 February 2004]

PWG Approved 16 Oct 2003 -- Release Date: 22 Jan 2004

In 5.2, notes for table 5-1:

1. L1 entry may be requested within ASPM protocol if both components on the link support and are enabled for ASPM L1~~, however its support is optional~~.

In 5.4.1.2, add:

When supported, L1 entry is disabled by default in the ASPM Control configuration field. Software must enable ASPM L1 on the downstream component only if it is supported by both components on a Link. Software must sequence the enabling and disabling of ASPM L1 such that the upstream component is enabled before the downstream component and disabled after the downstream component.

Section 5.4.1.2.1 Entry to L1 State:

An Upstream Port on a Switch or Endpoint enabled for L1 ASPM entry may initiate entry into the L1 Link state.

IMPLEMENTATION NOTE

Initiating L1

This specification does not dictate when an Endpoint must initiate a transition to the L1 state~~ on its Transmit Lanes~~. The interoperable mechanisms for transitioning into and out of L1 are defined within this specification, however the specific ASPM policy governing when to transition into L1 is left to the implementer.

One possible approach would be for the Downstream device to initiate a transition to the L1 state once the device has both its Receiver and Transmitter in the L0s state (RxL0s and TxL0s) for a set amount of time.

Three power management Messages provide support for ASPM of the L1 state:

- PM_ Active_State_Request_L1 (DLLP)
- PM_ Request_Ack (DLLP)
- PM_Active_State_Nak (TLP)

~~Endpoints~~ Downstream components enabled for ~~that have their~~ ASPM L1 entry ~~enabled~~ negotiate for ~~the resultant L-state~~ L1 entry with the upstream component on the ~~upstream end of the~~ Link. ~~If the Endpoint receives a negative acknowledgement in response to its issuance of a PM_Active_State_Request_L1 DLLP, the Endpoint must enter the L0s state as soon as possible. [Footnote: Assuming L0s is enabled and that the conditions for L0s entry are met.] Note that the component on the upstream side of the Link may not support L1 ASPM, or it may be disabled and so could legitimately respond to the L1 entry request with a negative acknowledgement.~~

A ~~Root Complex Root Port, or Switch~~ Downstream Port must accept a request to enter ~~a low power~~ L1 ~~state~~ if all of the following conditions are true:

- The Port supports ASPM L1 entry, and ASPM L1 entry is enabled[Footnote: Software must enable ASPM L1 for the Downstream Component only if it is also enabled for the Upstream Component.]~~.~~
- No TLP is scheduled for transmission
- No Ack or Nak DLLP is scheduled for transmission

A Switch Upstream Port may request L1 entry on its Link provided all of the following conditions are true:

- The Upstream Port supports ASPM L1 entry and it is enabled
- All of the Switch's Downstream Port Links are in the L1 state (or deeper)
- No pending TLPs to transmit
- No pending DLLPs to transmit
- The Upstream Port's Receiver is ~~Lanes are~~ idle for an implementation specific set amount of time

~~If the Switch's Upstream Port receives a negative acknowledgement in response to its issuance of a PM_Active_State_Request_L1 DLLP, then the Switch's Upstream Port Transmit Lanes must instead transition to the L0s state as soon as possible. [Footnote: Assuming L0s is enabled and that the conditions for L0s entry are met.]~~

...

In 7.8.7 Link Control Register (Offset 10h):

...

Table 7-15: Link Control Register

Bit Location	Register Description	Attributes

Bit Location	Register Description	Attributes
1:0	**Active State Power Management (ASPM) Control** – This field controls the level of ASPM supported on the given PCI Express Link. Defined encodings are: 00b Disabled 01b L0s Entry Enabled 10b L1 Entry Enabled 11b L0s and L1 Entry Enabled Note: "L0s Entry Enabled" indicates the Transmitter entering L0s is supported. The Receiver must be capable of entering L0s even when the field is disabled (00b). Default value of this field is (00b) or (01b) depending on form factor. ASPM L1 must be enabled by software in the upstream component on a link prior to enabling ASPM L1 in the downstream component on that link. When disabling ASPM L1, software must disable ASPM L1 in the downstream component on a link prior to disabling ASPM L1 in the upstream component on that link. ASPM L1 must only be enabled on the downstream component if both components on a link support ASPM L1.	RW
	...	

C9. Location of Set Slot Power Limit Message Data Fields

PWG Approved 16 Oct 2003 -- Release Date: 22 Jan 2004

In 2.2.8.5:

The Set_Slot_Power_Limit Message includes a one DW data payload. The data payload is copied from the Slot Capabilities register of the Downstream Port and is written into the Device Capabilities register of the Upstream Port on the other side of the Link. Bits ~~9:8~~1:0 of Byte one of the data payload map to the Slot Power Limit Scale field and bits 7:0 of Byte zero map to the Slot Power Limit Value field. Bits ~~31:10~~ 7:0 of Byte three, 7:0 of Byte two, and 7:2 of Byte one of the data payload must be set to all 0's by the Transmitter and ignored by the Receiver. ...

C10. Root Complex Peer to Peer & Payload Splitting

PWG Approved 16 Oct 2003 -- Release Date: 22 Jan 2004

Although support for peer-to-peer transaction routing between root ports of a root complex is both optional and implementation dependent, a root complex that supports peer-to-peer routing of Vendor_Defined Messages between its root ports is subject to specific requirements for interoperability reasons.

In Section 1.3.1, add to the third bullet as follows:

- The capability to route peer-to-peer transactions between hierarchy domains through a Root Complex is optional and implementation dependent. For example, an implementation may incorporate a real or virtual Switch internally within the Root Complex to enable full peer-to-peer support in a software transparent way.

  Unlike the rules for a Switch, a Root Complex is generally permitted to split a packet into smaller packets when routing transactions peer-to-peer between hierarchy domains (except as noted below), e.g., split a single packet with a 256-byte payload into two packets of 128 bytes payload each. The resulting packets are subject to the normal packet formation rules contained in this specification (e.g., Max_Payload_Size, Read Completion Boundary, etc.). Component designers should note that splitting a packet into smaller packets may have negative performance consequences, especially for a transaction addressing a device behind a PCI Express to PCI/PCI-X bridge.

  Exception: A Root Complex that supports peer-to-peer routing of Vendor_Defined Messages is not permitted to split a Vendor_Defined Message packet into smaller packets except at 128 byte boundaries (i.e., all resulting packets except the last must be an integral multiple of 128 bytes in length) in order to retain the ability to forward the Message across a PCI Express to PCI/PCI-X Bridge. Refer to the *PCI Express to PCI/PCI-X Bridge Specification, Rev. 1.0* for additional information.

C11. Primary Bus Number Register

PWG Approved 16 Oct 2003 -- Release Date: 22 Jan 2004

Background: PCI Express functions are required to capture the Bus (and Device) Number supplied with all Type 0 Configuration Write requests that are completed by the function. Unlike PCI-X, this requirement also applies to functions with Type 1 (bridge-type) Configuration Space headers. Additionally, PCI Express does not support forwarding Configuration Requests upstream nor the generation of PCI/PCI-X Special Cycles. As a result, the Primary Bus Number register in PCI Express functions with Type 1 Configuration Space headers is not used by the function. Note that PCI Express to PCI/PCI-X Bridges follow the PCI-X model.

Add new Section 7.5.3.2 (displacing existing 7.5.3.2 to 7.5.3.3, etc.) after Section 7.5.3.1:

7.5.3.2 Primary Bus Number (Offset 18h)

Except as noted, this register is not used by PCI Express functions but must be implemented as read-write for compatibility with legacy software. PCI Express functions capture the Bus (and Device) Number as

described in Section 2.2.6. See PCI Express to PCI/PCI-X Bridge Specification, Rev 1.0 for exceptions to this requirement.

C12. Assorted Clarifications & Corrections

PWG Approved 16 Oct 2003 -- Release Date: 22 Jan 2004

In 6.5.2, add as shown:

6.5.2. Initiation and Propagation of Locked Transactions -

...

- The Unlock Message is sent from the Root Complex ...
  - Any device which ...
- Any violation of the rules for initiation and propagation of locked transactions can result in undefined device and/or system behavior

---

In Table 2-18, Section 2.2.8.6, edit as shown:

Table 2-18: Vendor_Defined Messages

Name	Code[7:0]	Routing r[2:0]	Support				Req ID	Description/Comments
			RC	Ep	Sw	Br		
Vendor_Defined Type 0	0111 1110	000, 010, 011, 100	See note.				See note.	Triggers detection of UR by Completer~~Receiver~~ if not implemented.
Vendor_Defined Type 1	0111 1111	000, 010, 011, 100	See note.				See note.	Silently discarded by Completer~~Receiver~~ if not implemented.

Note: Implementation specific.

In Section 2.3.1. edit as shown:

- If the Request is a Message, and the Message Code specifies a value that is undefined, or that corresponds to a Message not supported by the device, ~~an undefined or unsupported value~~ (other than Vendor_Defined Type 1 which is not treated as an error

– see Section 2.2.8.6), the Request is an Unsupported Request, and is reported according to Section 6.2

In Section 2.3.2. edit as shown:

- Completions with a Completion Status other than Successful Completion, or Configuration Request Retry Status (in response to Configuration Request only) must cause the Requester to:
  - Free Completion buffer space and other resources associated with the Request.
  - ~~Report the error according to the rules in Section 6.2.~~

...

- Completions with a Reserved Completion Status value are treated as if the Completion Status was Unsupported Request (UR)
  - ~~This is a reported error associated with the Receiving device/function, normally the same as the Requestor (see Section 6.2)~~
- Completions with a Completion Status of Unsupported Request or Completer Abort are reported using the conventional PCI reporting mechanisms (see Section 7.5.1.2.)
  - Note that the error condition that triggered the generation of such a Completion is reported by the Completer as described in Section 6.2

In Section 7.11.2, Table 7-35, edit as shown:

Table 7-35: Port VC Capability Register 1

Bit Location	Register Description	Attributes
	…	
9:8	**Reference Clock** – Indicates the reference clock for Virtual Channels that support time-based WRR Port Arbitration. This field is valid for RCRB and for Switch Ports; it is not valid for Root Ports ~~and~~, Endpoint devices, and Switches or Root Complexes not implementing WRR, and must be set to 0. Defined encodings are: 00b 100 ns reference clock 01b – 11b Reserved	RO
	…	

In Section 7.11.3, Table 7-36, edit as shown:

Table 7-36: Port VC Capability Register 2

Bit Location	Register Description	Attributes
7:0	**VC Arbitration Capability** – Indicates the types of VC Arbitration supported by the device for the LPVC group. This field is valid for all devices that report a Low Priority Extended VC Count greater than 0. For all other devices this field must contain a value of 00h. Each bit location within this field corresponds to a VC Arbitration capability defined below. When more than one bit in this field is set, it indicates that the Port can be configured to provide different VC arbitration services. Defined bit positions are: Bit 0 — Hardware fixed arbitration scheme, e.g., Round Robin Bit 1 — Weighted Round Robin (WRR) arbitration with 32 phases Bit 2 — WRR arbitration with 64 phases Bit 3 — WRR arbitration with 128 phases Bits 4-7 — Reserved	RO
	...	

In Section 5.2:

All main power supplies, component reference clocks, and components' internal PLLs must be active at all times during L0s. TLP and DLLP communication ~~over~~through a Transmitter~~Link~~ that is in L0s is prohibited. The L0s state is used exclusively for ASPM.

In Section 5.2, page 224, delete 4th paragraph as shown:

~~Once the PME_Turn_Off/PME_TO_Ack sequence is initiated, the downstream component is guaranteed that a Fundamental Reset will occur (with or without the removal of power). Exit from L2/L3 Ready back to L0 must only be initiated through the power controller, and must follow entry into D3cold or Fundamental Reset.~~

In Section 5.3.1.4. D3 State:

...

System software must allow a ~~The~~ minimum recovery time following a D3hot ? D0 transition ~~is~~ of at least 10 ms, prior to accessing the function. This recovery time may, for example, be used by the D3hot ? D0 transitioning component to bootstrap any of its component interfaces (e.g., from serial ROM) prior to being accessible. Attempts to target the function during the recovery time (including configuration request packets) will result in undefined behavior.

Background: PCI-Express spec requires a function to capture its device number. This is OK for an EP device since an EP device is the only device connected to the virtual PCI bus link. However, this can cause problem for the internal virtual PCI bus of a switch where multiple downstream p2p ports are connected to. In the switch case, CfgW0 alone is not sufficient to differentiate among the multiple downstream p2p ports and device number must be used. This is a chicken-and-egg problem as the downstream ports need to know their own device number before it can be latched.

On p56 lines 1 – 5, edit as shown:

Functions must capture the Bus and Device Numbers supplied with all Configuration Write Requests (Type 0) completed by the function and supply these numbers in the Bus and Device Number fields of the Requester ID for all Requests initiated by the device/function.

Exception: The assignment of Bus and Device Numbers to the logical devices within a Root Complex, ***and Device Numbers to the downstream ports within a switch,*** may be done in an implementation specific way.

Background: The new DLLP timeout mechanism (p.108) recommends that a retraining be performed if we don't receive any UpdateFC DLLP's for up to 200us. However, what if the other side of the link initially broadcasted infinite flow control credits for all three classes? In this situation, they are not required to send UpdateFC's (p.104), and this timeout mechanism could be triggered incorrectly. The wording says "where the timer is reset by the receipt of any Init or Update FCP", but if the other side is legitimately not sending UpdateFC's, you could end up with a lot of retraining on your hands.

Add bullet at end:

* upon timer expiration ...

* If an Infinite Credit advertisement has been made during initialization for all three Flow Control classes, this timeout mechanism must be disabled

---

In Section 2.2.8.3, Table 2-15, edit as shown:

Table 2-15: Error Signaling Messages

Name	Code[7:0]	Routing r[2:0]	Support RC	Support Ep	Support Sw	Support Br	Req ID	Description/Comments
ERR_COR	0011 0000	000	r	t	rt	t	BD BDF	This Message is issued when the component or device detects a correctable error on the PCI Express interface.

ERR_NONFATAL	0011 0001	000	r	t	rt	t	BD BDF	This Message is issued when the component or device detects a Non-fatal, uncorrectable error on the PCI Express interface.
ERR_FATAL	0011 0011	000	r	t	rt	t	BD BDF	This Message is issued when the component or device detects a Fatal, uncorrectable error on the PCI Express interface.

In Section 2.3.1.1, edit as shown:

- Requests which do cross the address boundaries at integer multiples of RCB bytes may be completed using more than one Completion, but the data must not be fragmented except along the following address boundaries~~.~~:

...

In Section 5.3.1.2. edit as shown:

D1 support is optional. While in the D1 state, a function must not initiate any Request TLPs on the Link with the exception of a PME Message as defined in Section 5.3.3. ...

In Section 5.3.1.3. edit as shown:

D2 support is optional. While in the D2 state, a function must not initiate any Request TLPs on the Link with the exception of a PME Message as defined in Section 5.3.3. ...

In Section 5.3.1.4.2. edit as shown:

A function transitions to the D3cold state when its main power is removed. ...

C13. Command Complete Interrupt

SWG Approved 29 Oct 2003 -- Release Date: 22 Jan 2004

6.7.3.7. Command Completed Registers

Command Completed – This bit is set when the Hot-Plug Controller completes an issued command and is ready to accept the next command. The Command Completed status bit is set as an indication to host software

that the Hot-Plug Controller has processed the previous command and is ready to receive the next command; it provides no guarantee that the action corresponding to the command is complete.

Command Completed Interrupt Enable – This bit when set enables the generation of a Hot-Plug interrupt when a command is completed by the Hot-Plug control logic.

PCI Express Hot-Plug Controllers are required to implement the Command Completed register bits regardless of the form factor.

The Command Completed register bits apply~~ies~~ only to commands issued by software to control the Attention Indicator, Power Indicator or power controller. However, writes to other parts of the Slot Control Register would invariably end up writing to the indicators, power controller fields; Hence, any write transaction that targets any portion of the Slot Control Register is considered a command and if enabled, must result in a command complete interrupt. However, a write that disables the command complete interrupt is an exception to the above rule since such a write must not result in a command complete interrupt.

A single write to the Slot Control register is considered to be a single command, and hence receives a single command complete, even if the write affects more than one field in the Slot Control Register. Reads of the Slot Control register must reflect the value from the latest write, even if the corresponding command action is not complete, unless software issues a write without waiting for the previous command to complete in which case the read value is undefined.

The software must wait for confirmation of command completion (notification via Command Completed interrupt or polling Command Completed register) before issuing the next command. However, if the Command Completed register is not set 1 second after the command is issued, ~~the host~~ software is allowed to repeat the command or to issue the next command.

If software issues a write before the controller has completed processing of the previous command and before the 1 second time limit has expired, the controller is permitted to either accept or discard the write. Regardless, such a write is considered a programming error, and could result in a discrepancy between the Slot Control register and the hot plug element state. In order to recover from such a programming error and return the controller to a consistent state, software must issue a write to the Slot Control register which conforms to the command completion rules.

Software is expected to clear the command completed event by writing a 1 to the command completed bit. However, the controller must continue to process subsequent commands normally even if the software fails to clear the command completed status bit; software must not expect further command complete interrupts unless it has cleared the command completed status bit.

~~IMPLEMENTATION NOTE~~

~~When Does a Hot-Plug Controller Set the Command Completed Register?~~

~~A command completed notification is an indication to host software that the Hot-Plug Controller has processed the previous command and is ready to receive the next command; it provides no guarantee that the action corresponding to the command is complete.~~

~~Similarly, a read to the Slot Status register following a write will always return the value fromthe latest write, although the corresponding command action may not be complete.~~

C14. Differential Receiver Detect

EWG Approved 22 Jan 2004

Section 4.3.1.8: *At the end of this section, insert this implementation note:*

Implementation Note: Differential Receiver Detect

If an implementation chooses to transition from Detect to Polling based on electrical idle being broken prior to performing a receiver detect sequence, an unreliable link could be formed; due to the possibility that there may not be a low impedance termination resistor on both Rx differential conductors, which make up the differential pair.

If the Receiver Detection circuit performs the detect sequence on each conductor of the differential pair (both D+ and D-) and detects a load impedance greater than ZRX-DC on either conductor, the Receiver Detection circuit shall interpret this as no termination load present and respond as if neither load was present.

In this revision of this specification, it is not required that the detect sequence be performed on both conductors of a differential pair.

Future revisions of this specification may require the successful detection of a low impedance termination resistor on both differential pairs prior to transitioning to Polling.

C15. Crosslink Initialization

EWG Approved 22 Jan 2004

Section 4.2.6.3.1.1 [7th main bullet, page 176, starting at line 23] Replace:

Optionally, if crosslinks are supported and all Downstream Lanes initially receive two consecutive TS1 ordered sets with a Link number different than PAD (K23.7) and a Lane Number set to PAD, the Downstream Lanes are now designated as Upstream Lanes and a new random cross Link timeout is chosen (see Tcrosslink in Table 4-5). The next state is Configuration.Linkwidth.Start as Upstream Lanes.

With:

Optionally, if crosslinks are supported then all Downstream Lanes that detected a receiver during Detect must first transmit 16 - 32 TS1s with a non PAD Link number and PAD Lane number and after this occurs if any Downstream Lanes receive two consecutive TS1 ordered sets with a Link number different than PAD (K23.7) and a Lane Number set to PAD, the Downstream Lanes are now designated as Upstream Lanes and a new random cross Link timeout is chosen (see $T_{crosslink}$ in Table 4.5). The next state is Configuration.Linkwidth.Start as Upstream Lanes.

Section 4.2.6.3.1.2 [7th main bullet, page 177, starting at line 28] Replace:

Optionally, if crosslinks are supported and any Upstream Lanes first receive two consecutive TS1 ordered sets with Link and Lane numbers set to PAD (K23.7), then:

With:

Optionally, if crosslinks are supported then all Upstream Lanes that detected a receiver during Detect must first transmit 16 – 32 TS1s with a PAD Link number and PAD Lane number and after this occurs and if any

Upstream Lanes first receive two consecutive TS1 ordered sets with Link and Lane numbers set to PAD (K23.7), then:

C16. PME_Turn_Off

PWG Approved 4 Mar 2004 -- Release Date: 12 Jul 2004

In Section 5.2 Link State Power Management, edit as shown (note that this section includes changes from errata C12):

...

- L2/L3 Ready – Staging point for L2 or L3

  L2/L3 Ready transition protocol support is required

  ~~The L2/L3 Ready state is related to PCI-PM D-state transitions.~~ L2/L3 Ready is a ~~transitional~~ pseudo-state (~~not~~ corresponding ~~directly to the state of~~ to the LTSSM L2 state) that a given Link enters into when ~~the platform is actively executing the process to remove~~ preparing for the removal of power and clocks from the downstream component or both attached components. ...

  The L2/L3 Ready state entry transition process must begin as soon as possible following the acknowledgment of a PME_Turn_Off Message, (i.e., the injection of a PME_TO_Ack TLP). The Downstream component initiates L2/L3 Ready entry by injecting a PM_Enter_L23 DLLP onto its Transmit Port. Refer to Section 5.6 for further detail on power management system Messages.

  TLP and DLLP communication over a Link that is in L2/L3 Ready is not possible.

  Note: Exit from L2/L3 ready back to L0 will be through intermediate LTSSM states. Refer to Chapter 4 for detailed information.

- L2 – Auxiliary powered Link deep energy saving state.

  ...

- L3 – Link Off state.

  Zero power state.

- LDn – A transitional Link Down pseudo-state prior to L0

  This pseudo-state is associated with the LTSSM states Detect, Polling and Configuration, and, when applicable, Disabled, Loopback and Hot Reset.

...

Figure 5-1 highlights the legitimate L-state transitions that may occur during the course of Link operation.

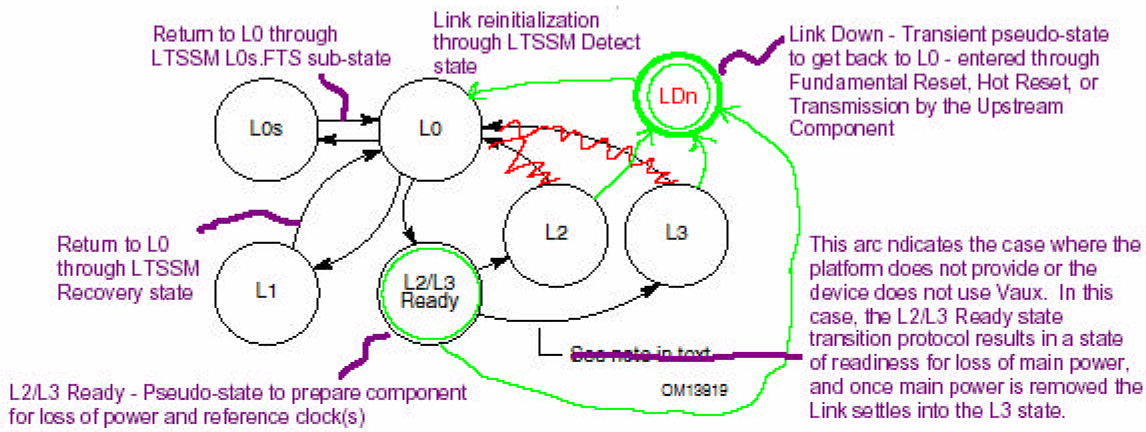


Figure 5-1: Link Power Management State Flow Diagram~~Transitions~~ *<edit figure as shown>*

Note that these states and state transitions do not correspond directly to the actions of the Physical Layer LTSSM. For example, the LTSSM is typically powered by main power (not Vaux), and so in both the L2 and L3 states will be unpowered. ~~The arc noted in Figure 5-1 indicates the case where the platform does not provide Vaux. In this case, the L2/L3 Ready state transition protocol results in a state of readiness for loss of main power, and once main power is removed the Link settles into the L3 state.~~

~~Link PM transitions from any L-state to any other L-state must pass through the L0 state, during the transition process with the exception of the L2/L3 Ready to L2 or L3 transitions. In this case, the Link transitions from L2/L3 Ready directly to either L2 or L3 when main power to the component is removed. (This follows along with a corresponding component's D-state transition from $D3_{hot}$ to $D3_{cold}$.)~~

The following example sequence, leading up to entering a system sleep state, illustrates the multi-step Link state transition process:

1. System Software directs all functions of a Downstream component to $D3_{hot}$.
2. The Downstream component then initiates the transition of the Link to L1 as required.
3. System Software then causes the Root Complex to broadcast the PME_Turn_Off Message in preparation for removing the main power source.
4. This Message causes the subject Link to transition back to L0 in order to send it, and to enable the Downstream component to respond with PME_TO_Ack.
5. After the PME_TO_Ack is sent, the Downstream component then initiates the L2/L3 Ready transition protocol.

L0 ? L1 ? L0 ? L2/L3 Ready

Table 5-1 summarizes each L-state, describing when they are used, and the PCI Express platform, and PCI Express component behaviors that correspond to each of them.

A "Yes" entry indicates that support is required (unless otherwise noted). "On" and "Off" entries indicate the required clocking and power delivery. "On/Off" indicates an optional design choice.

Table 5-1: Summary of PCI Express Link Power Management States <*Note relationship w/ECR #30 (not incl here)*>

	L-State Description	Used by S/W Directed PM	Used by ASPM	Platform Reference Clocks	Platform Main Power	Component Internal PLL	Platform Vaux
L0	Fully active Link	Yes (D0)	Yes (D0)	On	On	On	On/Off
L0s	Standby state	No	Yes [1] (D0)	On	On	On	On/Off
L1	Lower power standby	Yes (D1-D3hot)	Yes[2] (opt., D0)	On	On	On/Off [3]	On/Off
L2/L3 Ready (pseudo-state)	Staging point for power removal	Yes [4]	No	On	On	On/Off	On/Off
L2	Low power sleep state (all clocks, main power off)	Yes [5]	No	Off	Off	Off	On [6]
L3	Off (zero power)	n/a	n/a	Off	Off	Off	Off
LDn (pseudo-state)	Transitional state preceding L0	Yes	N/A	On	On	On/Off	On/Off

Notes:

1. L0s exit latency will be greatest in Link configurations characterized by independent reference clock inputs for components connected to opposite ends of a given Link (vs. a common, distributed reference clock).
2. L1 entry may be requested within ASPM protocol, however its support is optional.
3. L1 exit latency will be greatest for components that internally shut off their PLLs during this state.
4. L2/L3 Ready entry sequence is initiated at the completion of the PME_Turn_Off/PME_TO_Ack protocol handshake . It is not directly affiliated with a D-State transition, or a transition in accordance with ASPM policies and procedures.
5. Depending upon the platform implementation, the system's sleep state may use ~~utilize~~ the L2 state, ~~or~~ transition to ~~being~~ fully off (L3), or it may leave links in the L2/L3 Ready state. L2/L3 Ready state transition protocol is initiated by the Downstream component following reception and TLP acknowledgement of the PME_Turn_Off TLP Message. While platform support for an L2 sleep state configuration is optional (i.e., support for Vaux delivery), PCI Express component protocol support for transitioning the Link to the L2/L3 Ready state is required.
6. L2 is distinguished from the L3 state only by the presence of Vaux. After the completion of the L2/L3 Ready state transition protocol and before main power has been removed, the Link has indicated its readiness for main power removal.

In Section 5.3.2.3. Entry into the L2/L3 Ready State:

...

- The time for L2/L3 Ready entry transition is indicated by the completion of the PME_Turn_Off/PME_TO_Ack handshake sequence. Any actions on the part of the Downstream component necessary to ready itself for loss of power must be completed prior to initiating the transition to L2/L3 Ready. Once all preparations for loss of power and clock are completed, L2/L3 Ready entry is initiated by the Downstream component by sending the PM_Enter_L23 DLLP upstream.

  ~~In contrast, the time for L1 entry transition is indicated by programming all of the Downstream component's function(s) to non-D0 states, or by ASPM policies. There are no preparations necessary before initiating a transition to L1.~~

- The Downstream component must be in $D3_{hot}$ prior to being transitioned into the L2/L3 Ready state, i.e., a PME_Turn_Off Message must never be sent unless all functions Downstream of its point of origin are currently in $D3_{hot}$.

  In contrast, a Downstream component initiating a transition to L1 would have always initially been in D0, and had just been reprogrammed to D1, D2, or $D3_{hot}$.

- L2/L3 Ready entry transition protocol uses the PM_Enter_L23 DLLP.

  Note that the PM_Enter_L23 DLLPs are sent continuously until an acknowledgement is received or power is removed.

In Section 5.3.3.2.1 PME Synchronization:

~~PCI Express devices need to be notified before their reference clock and main power may be removed so that they can prepare for that eventuality.~~ PCI Express-PM introduces a fence mechanism that serves to initiate the power removal sequence while also coordinating the behavior of the platform's power management controller and PME handling by PCI Express agents.

~~Potential race conditions exist where a Downstream agent, if not somehow coordinated with the platform's power manager, could initiate a PM_PME Message while the power manager was in the process of turning off the main power source to the Link Hierarchy. The net result of hitting this corner condition would be loss of the PME indication. The fence mechanism ensures this does not happen.~~

PME_Turn_Off Broadcast Message

Before main component power and reference clocks are turned off, the Root Complex or Switch Downstream Port must issue a broadcast Message that instructs all agents downstream of that point within the hierarchy to cease initiation of any subsequent PM_PME Messages, effective immediately upon receipt of the PME_Turn_Off Message.

Each PCI Express agent is required to respond with a TLP “acknowledgement” ~~Packet~~Message, PME_TO_Ack that is, as in the case of a PME Message, always routed upstream. In all cases, the PME_TO_Ack Message must terminate at the PME_Turn_Off Message’s point of origin.[4]

~~Note that PM_PME and PME_TO_Ack, like all other PCI Express Message packets, are handled as posted transactions. It is their posted transaction nature that ensures that any previously injected PM_PME Messages will be pushed ahead of the fence acknowledgement assuring full in-order delivery of any previously initiated PM_PME Messages before the “Turn off” acknowledgement ever reaches the initiator of the PME_Turn_Off Message.~~

~~For the case where a PME_Turn_Off Message is initiated upstream of a PCI Express~~ A Switch~~, the PCI Express Switch’s Upstream Port~~ must report an “aggregate” acknowledgement only after having received PME_TO_Ack Messages ~~packets~~ from each of ~~their~~its Downstream Ports ~~individually~~. Once a PME_TO_Ack Message~~Packet~~ has arrived on ~~all~~ each Downstream Port~~s~~, the Switch must then send~~s~~ a PME_TO_Ack packet on its Upstream Port. The occurrence of any one of the following must reset the aggregation mechanism: the transmission of the PME_TO_Ack message from the upstream port, the receipt of any TLP at the upstream port, the removal of main power to the Switch, or fundamental reset.

All ~~PCI Express~~ Endpoints ~~components~~ must accept[~~5~~] and acknowledge the PME_Turn_Off ~~Packet~~Message from within the D3hot State. Once an Endpoint has sent a PME_TO_Ack Message~~Packet on its Transmit Link~~, it must then prepare for removal of its power and reference clocks by initiating a transition to the L2/L3 Ready state.

A Switch must ~~also~~ transition its Upstream Link to the L2/L3 Ready state ~~in the same manner as described in the previous paragraph for Endpoints. However, the Switch initiates this transition only~~ after all of its Downstream Ports have entered L2/L3 Ready state.

The Links attached to the originator of the PME_Turn_Off Message are the last to assume the L2/L3 Ready state. This state transition serves as an indication to the power delivery manager[6] that all Links within that portion of the PCI Express hierarchy have~~: S~~successfully retired all in flight PME Messages to the point of PME_Turn_Off Message origin and have ~~P~~performed any necessary local conditioning in preparation for power removal.

In order to avoid deadlock in the case where one or more devices do not respond with a PME_TO_Ack Message and then put their links into the L2/L3 Ready state, the power manager must implement a timeout after waiting for a certain amount of time, after which it proceeds as if the Message had been received and all links put into the L2/L3 Ready state. The recommended limit for this timer is in the range of 1ms to 10ms.

The power delivery manager must wait a minimum of 100 ns after observing all Links corresponding to the point of origin of the PME_Turn_Off Message enter L2/L3 Ready before removing the components’ reference clock and main power. This requirement does not apply in the case where the above mentioned timer triggers.

[4] Point of origin for the PME_Turn_Off Message could be all of the Root Ports for a given Root Complex (full platform sleep state transition), an individual Root Port, or a Switch Downstream Port.

~~[5] FC credits permitting.~~

[6] Power delivery control within this context relates to control over the entire PCI Express Link hierarchy, or over a subset of PCI Express Links ranging down to a single PCI Express Link and associated Endpoint~~s~~ for sub hierarchies supporting independently managed power and clock distribution.

IMPLEMENTATION NOTE

PME_TO_Ack Message Proxy by Switch Devices

One of the PME_Turn_Off/PME_TO_Ack handshake's key roles is to ensure that all in flight PME Messages are flushed from the PCI Express fabric prior to sleep state power removal. This is guaranteed to occur because PME Messages and the PME_TO_Ack Messages both use the posted request queue within ~~vc~~VC0 and so all previously injected PME Messages will be made visible to the system before the PME_TO_Ack is received at the Root Complex. Once all Downstream Ports of the Root Complex receive a PME_TO_Ack Message the Root Complex can then signal the power manager that it is safe to remove power without loss of any PME Messages.

...

~~IMPLEMENTATION NOTE~~

~~PME_TO_Ack Deadlock Avoidance~~

~~As specified earlier, any device that detects a PME_Turn_Off Message must reply with a PME_TO_Ack Message. However, system behavior must not depend on the correct behavior of any single device. In order to avoid deadlock in the case where one or more devices do not respond with a PME_TO_Ack Message, the power manager must not depend on the acceptance of a PME_TO_Ack Message. For example, the power manager may timeout after waiting for the PME_TO_Ack Message for a given time, after which it proceeds as if the Message was accepted.~~

In Section 5.3.3.4. PME Rules:

- All ~~PCI Express components supporting PCI Express-PM~~ devices must implement the PCI-PM PMC and PMCSR registers in accordance with the PCI-PM specification. These registers reside in the PCI-PM compliant PCI Capability List format.
    - PME capable functions must implement the PME_Status bit, and underlying functional behavior, in their PMCSR configuration register.
    - When a function initiates Link wakeup, or issues a PM_PME Message, it must set its PME_Status bit.
- Switches must route a PM_PME received on any Downstream Port to their Upstream Port
- On receiving a PME_Turn_Off Message while in the $D3_{hot}$ state, the device must block the transmission of PM_PME Messages and transmit a PME_TO_Ack Message upstream. The component is permitted to send a PM_PME Message after the Link is returned to an L0 state through LDn. ~~PME capable agents must comply with PME_Turn_Off and PME_TO_Ack fence protocols~~

- Before a Link or a portion of a Hierarchy is transferred into a non-communicating state (i.e., a state it cannot issue a PM_PME Message from), a PME_Turn_Off Message must be broadcast Downstream.

C17. Link & VC Initialization

PWG Approved 4 mar 2004 -- Release Date: 12 Jul 2004

In Section 2.6.1.:

...

- A received TLP using a~~n~~ ~~uninitialized~~ VC that is not enabled is a Malformed TLP
  - VC0 is always enabled
  - For VCs 1-7, a VC is considered enabled when the corresponding VC Enable bit in the VC Resource Control Register has been set to 1b, and once FC negotiation for that VC has exited the FC_INIT1 state and progressed to the FC_INIT2 state (see Section 3.3)
  - This is a reported error associated with the Receiving Port (see Section 6.2)

Note that TLP transmission for any VC should be postponed until the FC_INIT2 state is exited, because otherwise there is a risk that the receiving component on the other side of the Link will discard the TLP on reception (VC0 – see Section 2.9.1) or handle the TLP as a Malformed TLP (VCs 1-7). For VCs 1-7, software must use the VC Negotiation Pending bit in the VC Resource Status Register to ensure that a VC is not used until negotiation has completed by exiting the FC_INIT2 state in both components on a link.

In Section 2.9.1, Transaction Layer Behavior in DL_Down Status:

DL_Down status indicates that there is no connection with another component on the Link, or that the connection with the other component has been lost and is not recoverable by the Physical or Data Link Layers. This section specifies the Transaction Layer's behavior when the Data Link Layer reports DL_Down status to the Transaction Layer, indicating that the Link is non-operational.

- For a Port with DL_Down status, the Transaction Layer is not required to accept received TLPs from the Data Link Layer, provided that these TLPs have not been acknowledged by the Data Link Layer. Such TLPs do not modify receive Flow Control credits.

...

In Section 2.9.2, Transaction Layer Behavior in DL_Up Status:

DL_Up status indicates that a connection has been established with another component on the associated Link. This section specifies the Transaction Layer's behavior when the Data Link Layer reports entry to the DL_Up status to the Transaction Layer, indicating that the Link is operational. ~~These behaviors relate to Slot Power Limit support.~~

- The Transaction layer of a Port with DL_Up Status must accept received TLPs that conform to the other rules of this specification

...

In Section 3.2.1. Data Link Control and Management State Machine Rules:

...

- DL_Init
  - While in DL_Init:
    - Initialize Flow Control for the default Virtual Channel, VC0, following the Flow Control initialization protocol described in Section 3.3
    - Report DL_Down status while in state FC_INIT1; DL_Up status in state FC_INIT2
    - The Data Link Layer of a Port with DL_Down status is permitted to discard any received TLPs provided that it does not acknowledge those TLPs by sending one or more Ack DLLPs
  - Exit to DL_Active if:
    - Flow Control initialization completes successfully, and the Physical Layer continues to report Physical LinkUp = 1
      - Terminate attempt to initialize Flow Control for VC0 and Exit to DL_Inactive if:
    - Physical Layer reports Physical LinkUp = 0

...

In 3.3.1 Flow Control Initialization State Machine Rules, edit as shown:

...

- If at any time during initialization for VCs 1-7 the VC ~~x (where x does not equal 0) VCx~~ is disabled, the flow control initialization process for the VC~~x~~ is terminated
- Rules for state FC_INIT1:
  - Entered when initialization of a VC (VCx) is required
    - Entrance to DL_Init state (VCx = VC0)
    - When a VC (VCx = VC1-7) is enabled by software (see Section 7.11)
  - While in FC_INIT1:
    - Transaction Layer must block transmission of TLPs using VCx
    - Transmit the following uninterrupted sequence of three successive InitFC1 DLLPs for VCx in the following pattern:
      - InitFC1 – P (first)
      - InitFC1 – NP (second)
      - InitFC1 – Cpl (third)
    - Repeat this InitFC1 DLLP transmission sequence as follows:

    - For VC0, transmit continuously at the maximum rate possible on the Link (resend timer value is 0)
    - For VCs other than VC0, repeat the sequence when no other TLPs or DLLPs are available for Transmission, but no less frequently than at an interval of 17 µs (-0%/+100%), measured from the start of transmission of the preceding sequence
  - Except as needed to ensure at least the required frequency of InitFC1 DLLP transmission, the Data Link Layer must not block other transmissions
    - Note that this includes all Physical Layer initiated transmissions (for example, ordered sets), Ack and Nak DLLPs (when applicable), and TLPs using VCs that have previously completed initialization (when applicable)
  - Process received InitFC1 and InitFC2 DLLPs:
    - Record the indicated FC unit values
    - Set Flag FI1 once FC unit values have been recorded for each of P, NP, and Cpl for VCx
- Exit to FC_INIT2 if:
  - Flag FI1 has been set indicating that FC unit values have been recorded for each of P, NP, and Cpl for VCx

❑ Rules for state FC_INIT2:

- While in FC_INIT2:
  - ~~Transmission of TLPs using VCx by the Transaction Layer is permitted~~
  - Transaction Layer must block transmission of TLPs using VCx
  - Transmit the following uninterrupted sequence of three successive InitFC2 DLLPs for VCx in the following pattern:
    - InitFC2 – P (first)
    - InitFC2 – NP (second)
    - InitFC2 – Cpl (third)
  - Repeat this InitFC2 DLLP transmission sequence as follows:
    - For VC0, transmit continuously at the maximum rate possible on the Link (resend timer value is 0)
    - For VCs other than VC0, repeat the sequence when no other TLPs or DLLPs are available for Transmission, but no less frequently than at an interval of 17 µs (-0%/+100%), measured from the start of transmission of the preceding sequence
  - Except as needed to ensure at least the required frequency of InitFC2 DLLP transmission, the Data Link Layer must not block other transmissions

        - Note that this includes all Physical Layer initiated transmissions (for example, ordered sets), Ack and Nak DLLPs (when applicable), and TLPs using VCs that have previously completed initialization (when applicable)
    - Process received InitFC1 and InitFC2 DLLPs:
        - Ignore the indicated FC unit values
        - Set flag FI2 on receipt of any InitFC2 DLLP for VCx
    - Set flag FI2 on receipt of any TLP on VCx, or any UpdateFC DLLP for VCx
- Signal completion and exit if:
    - Flag FI2 has been set

- Violations of Flow Control initialization protocol are Data Link Layer Protocol Errors (DLLPE). Checking for such errors in FC initialization protocol is optional. If checking is implemented, any detected error is a reported error associated with the Port (see Section 6.2)

...

In 7.11.8, VC Resource Status Register, Table 7-41, edit as shown:

Bit Location	Register Description	Attributes
	...	
1	**VC Negotiation Pending** – This bit indicates whether the Virtual Channel negotiation (initialization or disabling) is in pending state. This field is valid for all devices. When this bit is set by hardware, it indicates that the VC resource has not completed ~~is still in~~ the process of negotiation. This bit is cleared by hardware after the VC negotiation is complete (on exit from the FC_INIT2 state). ~~For a non-default Virtual Channel, software may use this bit when enabling or disabling the VC. For the default VC, this bit indicates the status of the process of Flow Control initialization.~~ Before using a Virtual Channel, software must check whether the VC Negotiation Pending fields for that Virtual Channel are cleared in both components on thea Link.	RO

C18. Switch Message Handling

PWG Approved 23 Feb 2004 -- Release Date: 12 Jul 2004

In Section 2.2.8.4, Table 2-16:

Table 2-16 Unlock Message

Name	Code[7:0]	Routing r[2:0]	Support				Req ID	Description/Comments
			RC	Ep	Sw	Br		
Unlock	0000 0000	011	t	r	tr	r	BD	Unlock Completer

In Section 2.2.8.6, Table 2-18:

Table 2-18: Vendor_Defined Messages

Name	Code[7:0]	Routing r[2:0]	Support				Req ID	Description/Comments
			RC	Ep	Sw	Br		
Vendor_Defined Type 0	0111 1110	000, 010, 011, 100	See ~~note~~ Note 1.				See ~~note~~ Note 2.	Triggers detection of UR by Receiver if not implemented.
Vendor_Defined Type 1	0111 1111	000, 010, 011, 100	See ~~note~~ Note 1.				See ~~note~~ Note 2.	Silently discarded by Receiver if not implemented.

Note 1: Transmission by Endpoint/Root Complex/Bridge is i~~I~~mplementation specific. Switches must forward received Messages using Routing[2:0] field values of 000, 010 & 011.

Note 2: Implementation specific.

C19. Register Value Descriptions

PWG Approved 11 Mar 2004 -- Release Date: 12 Jul 2004
SWWG Approved 14 Apr 2004 -- Release Date: 12 Jul 2004

In Section 5.4.1.3:

Table 5-8: Encoding of the Endpoint L0s Acceptable Latency Field

Field	Description

Field	Description
Endpoint L0s Acceptable Latency	000b – ~~Less than~~ Maximum of 64 ns 001b – ~~64 ns to less than~~ Maximum of 128 ns 010b – ~~128 ns to less than~~ Maximum of 256 ns 011b – ~~256 ns to less than~~ Maximum of 512 ns 100b – ~~512 ns to less than~~ Maximum of 1 µs 101b – ~~1 µs to less than~~ Maximum of 2 µs 110b – ~~2 µs to less than~~ Maximum of 4 µs 111b – ~~More than 4 µs~~No Limit

Table 5-9: Encoding of the Endpoint L1 Acceptable Latency Field

Field	Description
Endpoint L1 Acceptable Latency	000b – ~~Less than~~ Maximum of 1 µs 001b – ~~1 µs to less than~~ Maximum of 2 µs 010b – ~~2 µs to less than~~ Maximum of 4 µs 011b – ~~4 µs to less than~~ Maximum of 8 µs 100b – ~~8 µs to less than~~ Maximum of 16 µs 101b – ~~16 µs to less than~~ Maximum of 32 µs 110b – ~~32 µs to~~ Maximum of 64 µs 111b – ~~More than 64 µs~~No Limit

In Section 7.8.3. Device Capabilities Register (Offset 04h), Table 7-11:

Table 7-11: Device Capabilities Register

Bit Location	Register Description	Attributes
…		

Bit Location	Register Description	Attributes
8:6	**Endpoint L0s Acceptable Latency** – This field indicates the acceptable total latency that an Endpoint can withstand due to the transition from L0s state to the L0 state. It is essentially an indirect measure of the Endpoint's internal buffering. Power management software uses the reported L0s Acceptable Latency number to compare against the L0s exit latencies reported by all components comprising the data path from this Endpoint to the Root Complex Root Port to determine whether ASPM L0s entry can be used with no loss of performance. Defined encodings are: 000b ~~Less than~~ Maximum of 64 ns 001b ~~64 ns to less than~~ Maximum of 128 ns 010b ~~128 ns to less than~~ Maximum of 256 ns 011b ~~256 ns to less than~~ Maximum of 512 ns 100b ~~512 ns to less than~~ Maximum of 1 µs 101b ~~1 µs to less than~~ Maximum of 2 µs 110b ~~2 µs-~~ Maximum of 4 µs 111b ~~More than 4 µs~~No Limit For devices other than Endpoints, this field is Reserved and must be 000b.	RO

Bit Location	Register Description	Attributes
11:9	**Endpoint L1 Acceptable Latency** – This field indicates the acceptable latency that an Endpoint can withstand due to the transition from L1 state to the L0 state. It is essentially an indirect measure of the Endpoint's internal buffering. Power management software uses the reported L1 Acceptable Latency number to compare against the L1 Exit Latencies reported (see below) by all components comprising the data path from this Endpoint to the Root Complex Root Port to determine whether ASPM L1 entry can be used with no loss of performance. Defined encodings are: 000b ~~Less than~~ Maximum of 1µs 001b ~~1 µs to less than~~ Maximum of 2 µs 010b ~~2 µs to less than~~ Maximum of 4 µs 011b ~~4 µs to less than~~ Maximum of 8 µs 100b ~~8 µs to less than~~ Maximum of 16 µs 101b ~~16 µs to less than~~ Maximum of 32 µs 110b ~~32 µs-~~ Maximum of 64 µs 111b ~~More than 64 µs~~No Limit For devices other than Endpoints, this field is Reserved and must be 000b.	RO
…		

In Section 7.8.8. Link Status Register (Offset 12h), Table 7-16:

Table 7-16: Link Status Register

Bit Location	Register Description	Attributes
3:0	**Link Speed** – This field indicates the negotiated Link speed of the given PCI Express Link. Defined encodings are: 0001b 2.5 Gb/s PCI Express Link All other encodings are reserved. The value in this field is undefined when the Link is not up.	RO
9:4	**Negotiated Link Width** – This field indicates the negotiated width of the given PCI Express Link. Defined encodings are: 000001b X1 000010b X2 000100b X4 001000b X8 001100b X12 010000b X16 100000b X32 All other encodings are reserved. The value in this field is undefined when the Link is not up.	RO
	...	

In 7.6. PCI Power Management Capability Structure:

This structure is required for all PCI Express devices. This capability is defined in the *PCI Bus Power Management Interface Specification, Rev. 1.1*. The functionality associated with this structure is the same for PCI Express as it is for conventional PCI, and only the added requirements associated with PCI Express are included here. ~~Figure 7-7 details allocation of the PCI PM Capability structure register fields in a PCI Express Context.~~

PCI Express devices are required to support D0 and D3 device states (refer to Section 5.1.1); PCI-PCI Bridge structures representing PCI Express Ports as described in Section 7.1 are required to indicate PME Message passing capability due to the in-band nature of PME messaging for PCI Express.

The PME Status bit for the PCI-PCI Bridge structure representing PCI Express Ports, however, is only set when the PCI-PCI Bridge function is itself generating a PME. The PME Status bit is not set when the Bridge is propagating a PME Message but the PCI-PCI Bridge function itself is not internally generating a PME.

Figure 7-8 details allocation of register fields for Power Management Capabilities register; and Table 7-7 describes the added requirements~~establishes the mapping between PCI 2.3 and PCI Express~~ for this register.

~~<Figure>~~

~~**Figure 7-7: PCI Power Management Capability Structure**~~

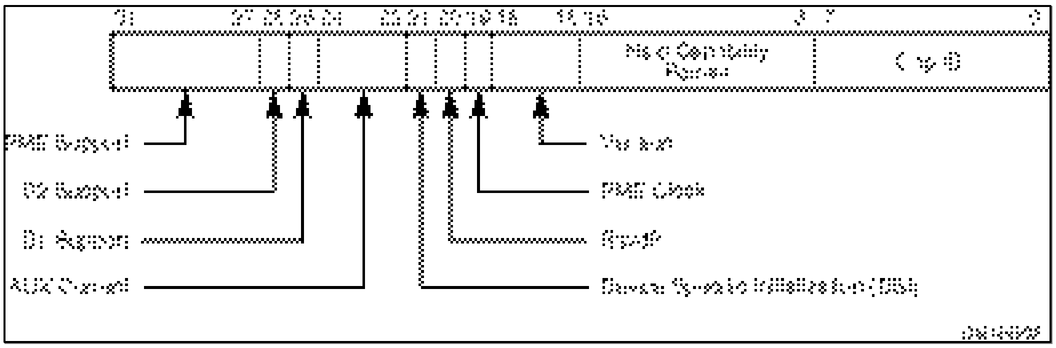


Figure 7-8: Power Management Capabilities Register

Table 7-7: Power Management Capabilities Register Added Requirements

Bit Location	Register Description	Attributes
~~7:0~~	~~**Capability ID** – Must be set to 01h~~	~~RO~~
~~15:8~~	~~**Next Capability Pointer**~~	~~RO~~
18:16	**Version** – Set to 02h for this version of the specification.	~~RO~~ unchanged
19	**PME Clock** – Does not apply to PCI Express. Must be hardwired to 0.	~~RO~~unchanged
~~21~~	~~**Device Specific Initialization**~~	~~RO~~
~~24:22~~	~~**AUX Current**~~	~~RO~~
~~25~~	~~**D1 Support**~~	~~RO~~
~~26~~	~~**D2 Support**~~	~~RO~~
31:27	**PME Support** – For a device, this 5-bit field indicates the power states in which the device may generate a PME. Bits 31, 30, and 27 must be set to 1b for PCI-PCI Bridge structures representing Ports on Root Complexes/Switches to indicate that the Bridge will forward PME Messages.	~~RO~~unchanged

Figure 7-9 details allocation of register fields for Power Management Status and Control register; and Table 7-8 describes the added requirements~~establishes the mapping between PCI 2.3 and PCI Express~~ for this register.

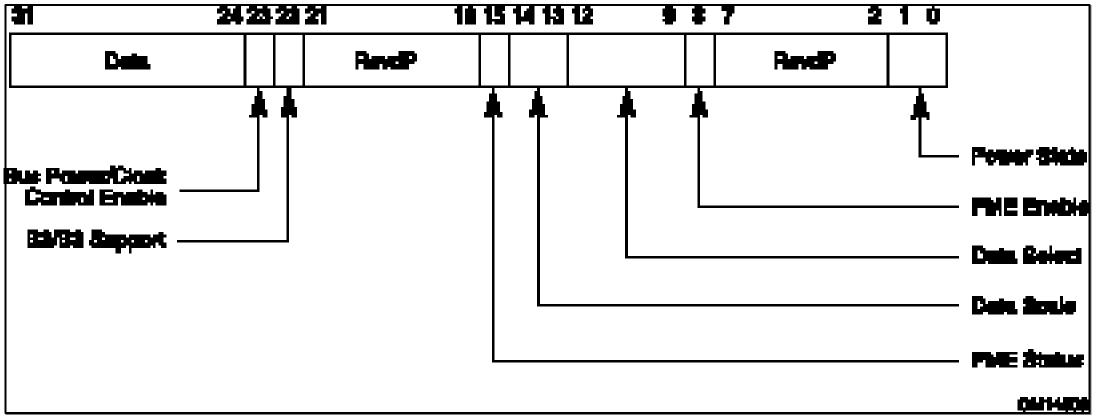


Figure 7-9: Power Management Status/Control Register

Table 7-8: Power Management Status/Control Register Added Requirements

Bit Location	Register Description	Attributes
~~1:0~~	~~**Power State**~~	~~RW~~
8	**PME Enable** – No added requirements Note: Devices that consume AUX power must preserve the value of this sticky register when AUX power is available. In such devices, this register value is not modified by hot, warm, or cold reset.	~~RWS~~unchanged
~~12:9~~	~~**Data Select**~~	~~RW~~
~~14:13~~	~~**Data Scale**~~	~~RO~~
15	**PME Status** – No added requirements Note: Devices that consume AUX power must preserve the value of this sticky register when AUX power is available. In such devices, this register value is not modified by hot, warm, or cold reset.	~~RW1CS~~unchanged
22	**B2/B3 Support** – Does not apply to PCI Express. Must be hardwired to 0.	~~RO~~unchanged
23	**Bus Power/Clock Control Enable** – Does not apply to PCI Express. Must be hardwired to 0.	~~RO~~unchanged
~~31:24~~	~~**Data**~~	~~RO~~

C20. Miscellaneous

PWG Approved 23 Feb 2004 -- Release Date: 12 Jul 2004

In 2.2.6.2. *Transaction Descriptor – Transaction ID Field :*

- Functions must capture the Bus and Device Numbers supplied with all Configuration Write Requests (Type 0) ...
- When generating Requests on their own behalf (for example, for error reporting), Switches must use the Requester ID associated with the primary side of the bridge logically associated with the Port (see Section 7.1) causing the Request generation.

In 2.3.1. Request Handling Rules:

...

- If the Request Type is not supported (by design or because of configuration settings) by the device, the Request is an Unsupported Request, and is reported according to Section 6.2
  - If the Request requires Completion, a Completion Status of UR is returned (see Section 2.2.9)

In 2.7.2.1. Error Forwarding Usage Model:

...

- Error Forwarding is u~~U~~sed for controlled propagation of errors through the system, system diagnostics, etc.
- Note that error Forwarding d~~D~~oes not cause Link Layer Retry – Poisoned TLPs will be retried only if there are transmission errors on PCI Express as determined by the TLP error detection mechanisms in the Data Link Layer.
  - The Poisoned TLP may ultimately cause the originator of the request to re-issue it (at the Transaction Layer or~~of~~ above) in the case of read operation or to take some other action. Such use of Error Forwarding information is beyond the scope of this specification.

In 3.5.3.1. LCRC and Sequence Number Rules (TLP Receiver):

...

- If the Physical Layer indicates a Receiver Error, discard any TLP currently being received and free any storage allocated for the TLP. Note that reporting such errors to software is done by the Physical Layer (and so are not reported by the Data Link Layer).
  - If a TLP was being received at the time the Receive Error was indicated and the NAK_SCHEDULED flag is clear,
    - ? schedule a Nak DLLP for transmission immediately
    - ? set the NAK_SCHEDULED flag

...

- ? if not equal, the TLP is corrupt - discard the TLP and free any storage allocated for the TLP
  - ¦ If the NAK_SCHEDULED flag is clear,
    - ? schedule a Nak DLLP for transmission immediately
    - ? set the NAK_SCHEDULED flag

  This is a reported error associated with the Port (see Section 6.2).

...

- o Otherwise, the TLP is out of sequence (indicating one or more lost TLPs):
  - ? if the NAK_SCHEDULED flag is clear,
    - ¦ schedule a Nak DLLP for transmission immediately
    - ¦ set the NAK_SCHEDULED flag
    - ¦ report TLP missing

  This is a reported error associated with the Port (see Section 6.2).

...

- ❑ ~~In addition to the other requirements for sending Ack DLLPs, a~~An Ack ~~or Nak~~ DLLP must be transmitted when all of the following conditions are true:
  - o The Data Link Control and Management State Machine is in the DL_Active state
  - o TLPs have been ~~accepted~~forwarded to the Receive Transaction Layer, but not yet acknowledged by sending an Ack DLLP
  - o The AckNak_LATENCY_TIMER reaches or exceeds the value specified in Table 3-5
  - o The NAK_SCHEDULED flag is clear

    Note: The AckNak_LATENCY_TIMER must be restarted from 0 each time an Ack or Nak DLLP is scheduled for transmission
- ❑ Data Link Layer Ack DLLPs may be scheduled for transmission~~Transmitted~~ more frequently than required

C21. Hot Plug

PWG Approved 22 Apr 2004 -- Release Date: 12 Jul 2004

In Document Organization (p.17):

... The mechanical chapters of the specification contain a definition of evolutionary PCI Express card edge connectors while the electrical chapters cover auxiliary signals, power delivery, and the ~~add-in card~~ adapter interconnect electrical budget.

...

Adapter Used generically to refer to an add-in card or module.

...

Slot Used generically to refer to an add-in card slot or module bay.

...

~~Standard Hot-Plug Controller (SHPC)~~ ~~A PCI Hot Plug Controler compliand with SHPC 1.0~~

...

2.2.8.7 Hot-Plug Signaling Messages

The ~~Hot~~hot-~~Plug~~ plug ~~Signaling~~ signaling ~~Messages~~ messages (Table 2-19) are virtual signals between Switches/Root Ports used to support ~~Hot~~hot-~~Plug~~ plug ~~Event~~ event signaling and devices on ~~cards~~ adapters that support Removal Request functionality (doorbell mechanism) on the ~~card~~adapter. For more information see Section 6.7.

...

Components must accept received ~~Hot~~hot-~~Plug~~ plug ~~Signaling~~ signaling ~~Messages~~ messages as defined in the Support column of Table 2-19 without indicating an error. However, only components which implement or support the associated action must process the received Message and take the corresponding action. Components which do not implement or support the associated action discard the received Message.

...

6.6. PCI Express Reset – Rules *<note – This section also modified in other errata (changes non-conflicting)>*

...

In all form factors and system hardware configurations, there must, at some level, be a hardware mechanism for setting or returning all Port states to the initial conditions specified in this document – this mechanism is called "Fundamental Reset". This mechanism can take the form of an auxiliary signal provided by the system to a component or ~~add-in card~~adapter, in which case the signal must be called PERST#, and must conform to the rules specified in Section 4.2.4.5.1. When PERST# is provided to a component or ~~add-in card~~adapter, this signal must be used by the component or ~~add-in card~~adapter to cause a Fundamental Reset. When PERST# is not provided to a component or ~~add-in card~~adapter, Fundamental Reset is generated autonomously by the component or an ~~add-in card~~adapter, and the details of how this is done are outside the scope of this document. If a Fundamental Reset is generated autonomously by the component or ~~add-in card~~adapter, and if power is supplied by the platform to the component/~~add-in card~~adapter, the component/~~add-in card~~adapter must generate a Fundamental Reset to itself if the supplied power goes outside of the limits specified for the form factor or system.

...

6.7 PCI Express ~~Native~~ Hot-Plug Support

The PCI Express architecture is designed to natively support both ~~Hot-Plug and hot-removal of devices. This section defines the standard usage model defined for all PCI Express form factors supporting Hot-Plug and hot removal of devices. This usage model provides the foundation for how indicators and push buttons should behave if implemented in a system. The definitions of indicators and push buttons apply to all PCI Express Hot-Plug models.~~hot-add and hot-removal ("hot-plug") of adapters and provides a "toolbox" of mechanisms that allow different user/operator models to be supported using a self-consistent infrastructure. This section defines the set of hot-plug mechanisms for PCI Express and specifies how the elements of hot-plug, such as indicators and push buttons, must behave if implemented in a system.

6.7.1 ~~PCI Express Hot-Plug Usage Model~~Elements of Hot-Plug

~~6.7.1.1. Usage Model Rationale~~

~~A standard usage model is beneficial to customers who buy systems with Hot-Plug slots because many customers utilize hardware and software from different vendors. A standard usage model allows customers to use the Hot-Plug slots on all of their systems without having to retrain operators. The PCI Express Hot-Plug standard usage model is derived from the standard usage model defined in the *PCI Standard Hot-Plug Controller and Subsystem Specification, Rev 1.0* and is identical from the user perspective; note that only slight changes were made in register definitions. Conformance to the standard usage model is required by all PCI Express form factors that implement Hot-Plug and use indicators and buttons. Specific requirements for individual elements of the standard usage model are form factor specific, and are defined in the individual form factor specifications.~~Table <6-5> lists the physical elements comprehended in PCI Express for support of hot-plug models. A PCI Express form-factor specification must define how these elements are used in that form-factor. For a given form-factor specification, it is possible that only some of the available hot-plug elements are required, or even that none of these elements are required. In all cases, the form-factor specification must define all assumptions and limitations placed on the system or the user by the choice of elements included. Silicon component implementations that are intended to be used only with selected form factors are permitted to support only those elements that are required by the associated form factor(s).

~~All PCI Express form factors that Support Hot-Plug/Remove are required to conform to the Standard Usage Model. Deviating from the Standard Usage Model causes the solution to be non-PCI Express compliant and will create issues that would not exist otherwise, such as:~~

- ~~User confusion~~
- ~~More extensive hardware testing~~
- ~~Functional incompatibilities with system software~~
- ~~Encountering untested paths in system software~~

~~6.7.1.2. Elements of the Standard Usage Model~~

Table 6-5: User Visible Elements of the Standard Usage Model

Element	Purpose
Indicators	Show~~s~~ the power and attention state of the slot
Manually-operated Retention Latch (MRL)	Holds ~~add-in cards~~ adapter in place

MRL Sensor	Allows the Port and system software to detect the MRL being opened
Electromechanical Interlock	Prevents removal of adapter ~~add-in cards~~ while slot is powered
Attention Button	Allows user to request ~~H~~hot-~~P~~plug operations
Software User Interface	Allows user to request ~~H~~hot-~~P~~plug operations
Slot Numbering	Provides visual identification of slots

6.7.1.2.1. Indicators

Two indicators are defined: the Power Indicator and the Attention Indicator. Each indicator is in one of three states: on, off, or blinking. Hot-plug system software has exclusive control of the indicator states by writing the command registers associated with the indicator (with one exception noted below).~~The Standard Usage Model defines two indicators; the Power Indicator and the Attention Indicator. The platform can provide the two indicators per slot or module bay and the indicators can be implemented either on the chassis or on the module. Each indicator is in one of three states: on, off, or blinking. Hot-Plug system software has exclusive control of the indicator states by writing the command registers associated with the indicator.~~

The indicator requirements must be included in all form-factor specifications. For a given form factor, the indicators may be required or optional or not applicable at all.

The ~~Hot~~hot-~~Plug-~~plug capable Port controls blink frequency, duty cycle, and phase of the indicators. Blinking indicators must operate at a frequency of between 1 ~~to~~ and 2 Hz~~,~~ ~~and~~ with a 50% (+/- 5%) duty cycle. Blinking indicators are not required to be synchronous ~~and~~ or in-phase between Ports.

~~Indicators must be placed in close proximity to their associated Hot-Plug slot if indicators are implemented on the chassis so that the association between the indicators and the Hot-Plug slot is clear.~~Indicators may be physically located on the chassis or on the adapter (see the associated form factor specification for Indicator location requirements). Regardless of the physical location, logical control of the indicators is by the downstream port of the upstream component on the link.

The Downstream Port must not change the state of an indicator unless commanded to do so by software, except for platforms capable of detecting stuck-on power faults (relevant only when a power controller is implemented). In the case of a stuck-on power fault, the platform is permitted to override the Downstream Port and force the Power Indicator to be on (as an indication that the adapter should not be removed). The handling by system software of stuck-on faults is optional and not described in this specification. Therefore, the platform vendor must ensure that this feature, if implemented, is addressed via other software, platform documentation, or by other means.

~~Both indicators are completely under the control of system software. The Switch device or Root Port never changes the state of an indicator in response to an event such as a power fault or unexpected MRL opening unless commanded to do so by software. An exception is granted to platforms capable of detecting stuck-on power faults. In the specific case of a stuck-on power fault, the platform is permitted to override the Switch device or Root Port and force the Power Indicator to be on (as an indication that the add-in card should not be removed). In all cases, the Ports internal state for the Power Indicator must match the software selected state. The handling by system software of stuck-on faults is optional and not described in this specification. Therefore, the platform vendor must ensure that this optional feature of the Standard Usage Model is addressed via other software, platform documentation, or by other means.~~

6.7.1.2.1.1. Attention Indicator

The Attention Indicator, which must be yellow or amber in color, ~~and~~ indicates that an operational problem exists or that the hot-p~~is yellow or amber in color and is used to indicate that an operational problem exists or that the Hot-P~~lug slot is being identified so that a human operator can locate it easily.

Table 6-6: Attention Indicator States

Indicator Appearance	Meaning
Off	Normal - Normal operation
On	Attention - Operational problem at this slot
Blinking	Locate - Slot is being identified at the user's request

Attention Indicator Off

~~When t~~The Attention Indicator ~~is off~~in the Off state~~, it means~~ indicates that neither the ~~add-in card~~adapter (if one is present) nor the ~~Hot~~hot-~~Plug~~ plug slot requires attention.

Attention Indicator On

~~When t~~The Attention Indicator ~~is on, it means~~in the On state indicates that an operational problem exists at the ~~card~~ adapter or slot.

An operational problem is a condition that prevents continued operation of an ~~add-in~~adapter~~card~~. The operating system or other system software determines whether a specific condition prevents continued operation of an adapter ~~add-in card~~ and whether lighting the Attention Indicator is appropriate. Examples of operational problems include problems related to external cabling, ~~add-in cards~~adapters, software drivers, and power faults. In general, ~~when~~ the Attention Indicator ~~is on, it means~~in the On state indicates that an operation was attempted and failed or that an unexpected event occurred.

The Attention Indicator is not used to report problems detected while validating the request for a ~~Hot~~hot-p~~P~~lug operation. Validation is a term applied to any check that system software performs to assure that the requested operation is viable, permitted, and will not cause problems. Examples of validation failures include denial of permission to perform a ~~H~~hot-p~~P~~lug operation, insufficient power budget, and other conditions that may be detected before ~~an operation begins~~a hot-plug request is accepted.

Attention Indicator Blinking

~~When the~~A blinking Attention Indicator ~~is blinking, it means~~indicates that system software is identifying this slot for a human operator to find. This behavior is controlled by a user (for example, from a software user interface or management tool).

6.7.1.2.1.2. Power Indicator

The Power Indicator, which must be ~~is~~ green in color~~, and is used to~~ indicates the power state of the slot. Table 6-7 lists the Power Indicator states.

Table 6-7: Power Indicator States

Indicator Appearance	Meaning

Off	Power Off - Insertion or removal of ~~add-in cards~~the adapter is permitted. ~~All supply voltages (except Vaux) have been removed from the slot if required for add-in card removal. Note that Vaux is removed when the MRL is open.~~
On	Power On - ~~The slot is powered on.~~ Insertion or removal of ~~add-in cards~~the adapter is not permitted.
Blinking	Power Transition - ~~The slot is in the process of powering up or down.~~ Hot-plug operation is in progress and i~~I~~nsertion or removal of ~~add-in cards~~the adapter is not permitted.

Power Indicator Off

~~When t~~The Power Indicator ~~is off~~in the Off state indicates~~, it means~~ that insertion or removal of ~~an add-in card~~the adapter is permitted. Main power to the slot is off if required by the form factor.

Note that, depending on the form factor, other power/signals may remain on even when main power is off and the Power Indicator is off. In ~~A~~an example ~~of main power removal is~~ using the PCI Express card form factor~~, I~~ if the platform provides Vaux to ~~H~~hot-~~P~~plug slots and the MRL is closed, any signals switched by the MRL are connected to the slot even when the Power Indicator is off. Signals switched by the MRL are disconnected when the MRL is opened. System software must cause a slot's Power Indicator to be turned off when the slot is not powered and/or it is permissible to insert or remove ~~add-in cards~~an adapter. See the appropriate ~~electromechanical specifications for form factor~~form factor specification for details.

Power Indicator On

~~When t~~The Power Indicator ~~is on~~in the On state indicates that the hot plug operation is complete and~~, it means~~ that main power to the slot is ~~on~~ On and that insertion or removal of ~~an~~the ~~add-in card~~adapter is not permitted.

Power Indicator Blinking

~~When the~~A blinking Power Indicator ~~is blinking, it means~~indicates that the slot is powering up or powering down and that insertion or removal of ~~an add-in card~~the adapter is not permitted.

~~A~~ The blinking Power Indicator also provides visual feedback to the ~~human~~ operator when the Attention Button is pressed or when a hot-plug operation is initiated through the hot-plug software interface.

6.7.1.2.2. Manually-operated Retention Latch (MRL)

An MRL is a manually-operated retention mechanism that holds an ~~add-in card~~adapter in the slot and prevents the user from removing the ~~card~~device. The MRL rigidly holds the ~~card~~ adapter in the slot so that cables may be attached without the risk of creating intermittent contact. MRLs that hold down two or more ~~add-in card~~adapter~~s~~s simultaneously are permitted in Platforms that do not provide MRL Sensors.

6.7.1.2.3. MRL Sensor

The MRL Sensor is a switch, optical device, or other type of sensor that reports the position of a slot's MRL to the downstream ~~Port~~port. The MRL Sensor reports closed when the MRL is fully closed and open at all other times (that is, if the MRL is fully open ~~and~~ or in an intermediate position~~s~~).

If a power controller is implemented for the slot, the slot main power ~~Signals switched by the MRL (such as Vaux and SMBus)~~ must be automatically removed from the slot when the MRL Sensor indicates that the MRL is open. If signals such as Vaux and SMBus are switched by the MRL, then these signals must be automatically removed from the slot when the MRL Sensor indicates that the MRL is open and must be

restored to the slot when the MRL Sensor indicates that MRL has closed again. Refer to the appropriate form factor specification to identify the signals, if any, switched by the MRL.

Note that the Hot-Plug Controller does not autonomously change the state of either the Power Indicator or the Attention Indicator based on MRL sensor changes.

~~The MRL Sensor allows the Port to monitor the position of the MRL and therefore allows the Port to detect unexpected openings of the MRL. When an unexpected opening of the MRL associated with a slot is detected, the Port changes the state of that slot to disabled and notifies system software. The Port does not autonomously change the state of either the Power Indicator or the Attention Indicator.~~

~~In the absence of an MRL sensor, staggered presence detect pins could be used to handle the switched signals. In this case, when the presence pins break contact, the switched signals will be automatically removed from the slot.~~

IMPLEMENTATION NOTE

MRL Sensor Handling

In the absence of an MRL sensor, for some form factors, staggered presence detect pins may be used to handle the switched signals. In this case, when the presence pins break contact, the switched signals will be automatically removed from the slot.

If an MRL Sensor is implemented without a corresponding MRL Sensor input on the PCI Express Hot-Plug Controller, it is recommended that the MRL Sensor be routed to the power fault input of the Hot-Plug Controller. This allows an active ~~PCI Express card or module~~adapter to be powered off when the MRL is opened.

6.7.1.2.4. Electromechanical Interlock

An electromechanical interlock is a mechanism for physically locking the ~~add-in card~~adapter or MRL in place until the system software and Port releases it. ~~Implementation of the interlock is optional. There is no mechanism in the programming interface for explicit control of electromechanical interlocks. The Standard Usage Model assumes that if electromechanical interlocks are implemented, they are controlled by the same Port output signal that enables main power to the slot.~~ Requirements for the electromechanical interlock are called out in the associated form factor specification. Systems may optionally expand control of interlocks to provide physical security of the adapter~~add-in cards~~.

6.7.1.2.5. Attention Button

~~An~~ The Attention Button is a momentary-contact push button switch, located adjacent to each ~~Hot~~hot-~~Plug~~ plug slot or on ~~a module~~the adapter that is pressed by the user to initiate a ~~hot-insertion or a hot removal~~hot-plug operation at that slot. Regardless of the physical location of the button, the signal is processed and indicated to software by hot-plug hardware associated with the downstream port corresponding to the slot.

In form factors that electrically control the Attention Button by the chassis, the Attention Button must allow the user to initiate both hot add and hot remove operations regardless of the physical location of the button. In form factors that electrically control the Attention Button by the component on the adapter, the attention button can be used to initiate hot remove operations only.

If present, t~~T~~he Power Indicator provides visual feedback to the human operator (if the system software accepts the request initiated by the Attention Button) by blinking. Once the Power Indicator begins blinking, a 5-second abort interval exists during which a second depression of the Attention Button cancels the operation.

If an operation initiated by an Attention Button fails for any reason, it is recommended that system software present an error ~~M~~message explaining the failure via a software user interface or add the error ~~Message~~ message to a system log.

6.7.1.2.6. Software User Interface

System software provides a user interface that allows hot insertions and hot removals to be initiated and that allows occupied slots to be monitored. A detailed discussion of ~~H~~hot-~~P~~plug user interfaces is operating system specific and is therefore beyond the scope of this document.

On systems with multiple ~~H~~hot-~~P~~plug slots, the system software must allow the user to initiate operations at each slot independent of the states of all other slots. Therefore, the user is permitted to initiate a ~~H~~hot-~~P~~plug operation on one slot using either the software user interface or the Attention Button while a ~~H~~hot-~~P~~plug operation on another slot is in process, regardless of which interface was used to start the first operation.

6.7.1.2.7. Slot Numbering

A Physical Slot Identifier (as defined in PCI HP 1.1, Section 1.5) consists of an optional chassis number and the physical slot number of the ~~Hot-Plug~~ slot. The physical slot number is a chassis unique identifier for a slot. System software determines the physical slot number from registers in the Port. ~~The c~~Chassis number is 0 is reserved for the main chassis. The chassis number for other chassis must be a non-zero value obtained from a PCI-to-PCI Bridge's Chassis Number register (see PCI Bridge 1.1, Section 13.4).

Regardless of the form factor associated with each slot, each physical slot number must be unique within a chassis. ~~The Standard Usage Model also requires that each physical slot number is globally unique within a chassis.~~

~~6.7.2. Event Behavior~~

~~Depending on the power state of the Switch device or Root Port, a Hot-Plug removal event may generate either a system interrupt or a PME (see Table 6-8).~~

~~Table 6-8: Event Behavior~~

~~Event~~	~~Register Bit Set When Detected~~	~~Cleared by~~	~~Port Optionally Generates the Following When Event is Detected:~~
~~Presence Detect Change~~	~~Presence Detect Event Status~~	~~Writing a 1 to the detected bit~~	~~System Interrupt, PME~~
~~Attention Button Pressed~~	~~Attention Button Pressed Event~~	~~Writing a 1 to the detected bit~~	~~System Interrupt, PME~~
~~MRL Sensor Changed~~	~~MRL Sensor Change Detected Event~~	~~Writing a 1 to the detected bit~~	~~System Interrupt, PME~~
~~Power Fault~~	~~Power Fault Detected Event~~	~~Writing a 1 to the detected bit.~~	~~System Interrupt, PME~~

6.7.3. ~~Port~~ Registers Grouped by ~~Device~~ Hot-Plug Element Association

The registers described in this section are grouped by ~~device~~ hot-plug element to convey all registers associated with implementing each ~~device~~ element ~~in Ports~~. ~~These r~~Registers ~~are unique to~~ associated with each Downstream Port ~~Switch device or Root Port~~ implementing a ~~Hot~~hot-~~Plug~~ plug capable slot~~s, and~~ are located in the PCI Express Capabilities, Slot Capabilities, Slot Control, and Slot Status Registers in the PCI Express Capability structure (Section 7.8). Registers reporting the presence of hot-plug elements associated with the device on an adapter are located in the Device Capabilities Register (also in the PCI Express Capability structure).

6.7.3.1. Attention Button Registers

Attention Button Present (Slot Capabilities & Device Capabilities) – This bit indicates if an Attention Button is ~~implemented on~~electrically controlled by the chassis (Slot Capabilities) or by the adapter (Device Capabilities).

Attention Button Pressed (Slot Status) – This bit is set when ~~the~~ an Attention Button electrically controlled by the chassis is pressed. ~~This register is set by the debounced output of an Attention Button.~~ This bit is also set ~~by the Port~~ upon receipt~~ving the~~ of an Attention_Button_Pressed Message from the ~~end device~~adapter.

Attention Button Pressed Enable (Slot Control) – When set, ~~This~~ this bit ~~when set~~ enables software notification~~the generation of the hot-plug interrupt or a wakeup event~~ on an Attention Button Pressed event (see Section 6.7.7.3).

6.7.3.2. Attention Indicator Registers

Attention Indicator Present (Slot Capabilities & Device Capabilities) – This bit indicates if an Attention Indicator is ~~implemented on~~electrically controlled by the chassis (Slot Capabilities) or by~~on~~ the adapter (Device Capabilities).

Attention Indicator Control (Slot Control)– When written, sets an Attention Indicator electrically controlled by the chassis to the written state and~~.~~ ~~When read this register returns the current state of the Attention Indicator; when written the Attention Indicator is set to this state.~~ ~~If an Attention Indicator is implemented on the card, when written, the Port will~~ sends the appropriate Attention Indicator Message (determined by the decoding) to the device on the ~~card~~adapter.

6.7.3.3. Power Indicator Registers

Power Indicator Present (Slot Capabilities & Device Capabilities) – This bit indicates if a Power Indicator is ~~implemented on~~electrically controlled by the chassis (Slot Capabilities) or by the adapter (Device Capabilities).

Power Indicator Control (Slot Control) – W~~hen read this register returns the current state of the Power Indicator; w~~hen written, sets ~~the~~ a Power Indicator electrically controlled by the chassis ~~is set~~ to ~~this~~ the written state~~.~~ and ~~If a Power Indicator is implemented on the card, when written, the Port will~~ sends the appropriate Power Indicator Message (determined by the decoding) to the device on the ~~card~~adapter.

6.7.3.4. Power Controller Registers

Power Controller Present (Slot Capabilities) – This bit indicates if a Power Controller is implemented. ~~for this slot.~~

Power Controller Control (Slot Control) –Turns the Power Controller on or off according to the value written. ~~When read, this register returns the current state of the Power applied to the slot; when written, the Power Controller turns on or off power to slot.~~

Power Fault Detected (Slot Status) – This bit is set when ~~the Power Controller detects~~ a power fault is detected at the~~is~~ slot or the adapter.

Power Fault Detected Enable (Slot Control) – ~~This bit w~~When set, this bit enables software notification~~the generation of the hot-plug interrupt or a wakeup event~~ on a power fault event (see Section 6.7.7.3).

6.7.3.5. Presence Detect Registers

Presence Detect State (Slot Status) – This bit indicates the presence of an adapter ~~card~~ in the slot. ~~The bit reflects the Presence Detect status determined via in-band mechanism or via Present Detect pins such as defined in the PCI Express Card Electromechanical Specification. This register is required to be implemented on all Switch devices and Root Ports. The presence detect field for Switch devices or Root Ports not connected to slots should be hardwired to 1. This register is required if a slot implemented.~~

Presence Detect Changed ~~Event~~ (Slot Status) – This bit is set when a Presence Detect state change is detected.

Presence Detect Changed Enable (Slot Control) – ~~This bit w~~When set, this bit enables software notification ~~the generation of the hot-plug interrupt or a wakeup event~~ on a presence detect changed event (see Section 6.7.7.3)

<ed. note: The following text moved (not deleted)>

~~Tthe in-band presence detect mechanism requires that power be applied to a card for its presence to be detected. Consequently, form factors that require a power controller for Hot-Plug must implement a physical pin presence detect mechanism as defined in the *PCI Express Card Electromechanical Specification*.~~

~~IMPLEMENTATION NOTE~~

~~No Slot Power Controller~~

~~For slots that do not implement a power controller, software must ensure that system power planes are enabled to provide power to slots prior to reading presence detect state.~~

6.7.3.6. MRL Sensor Registers

MRL Sensor Present (Slot Capabilities) – This bit indicates if an MRL Sensor is implemented ~~on the chassis~~.

MRL Sensor Changed (Slot Status) – This bit is set when the value of the MRL Sensor state changes~~d~~.

MRL Sensor Changed Enable (Slot Control) – This bit when set enables ~~the generation of the hot-plug interrupt or a wakeup event~~software notification on a MRL Sensor changed event (see Section 6.7.7.3).

MRL Sensor State (Slot Status) – This register reports the status of the MRL Sensor if ~~it~~ one is implemented.

~~If switched signals are handled automatically by detecting an extraction event using the PRSNT#1 and PRSNT#2 pins, then MRL Sensor registers need not be implemented.~~

6.7.3.7. Command Completed Registers

Command Completed (Slot Status) – This bit is set when the Hot-Plug Controller completes an issued command and is ready to accept the next command. ~~The Command Completed status bit is set as an indication to host software that the Hot-Plug Controller has processed the previous command and is ready to receive the next command; it provides no guarantee that the action corresponding to the command is complete.~~

Command Completed Interrupt Enable (Slot Control) – This bit when set enables ~~the generation of a H~~h~~ot-P~~p~~lug interrupt~~software notification (see Section 6.7.7.3) when a command is completed by the ~~H~~hot-~~P~~plug control logic.

~~PCI Express Hot-Plug Controllers are required to implement the Command Completed register bits regardless of the form factor.~~

~~The Command Completed register bits apply only to commands issued by software to control the Attention Indicator, Power Indicator or power controller. However, writes to other parts of the Slot Control Register would invariably end up writing to the indicators, power controller fields; Hence, any write transaction that targets any portion of the Slot Control Register is considered a command and if enabled, must result in a command complete interrupt. However, a write that disables the command complete interrupt is an exception to the above rule since such a write must not result in a command complete interrupt.~~

~~A single write to the Slot Control register is considered to be a single command, and hence receives a single command complete, even if the write affects more than one field in the Slot Control Register. Reads of the Slot Control register must reflect the value from the latest write, even if the corresponding command action is not complete, unless software issues a write without waiting for the previous command to complete in which case the read value is undefined.~~

~~The software must wait for confirmation of command completion (notification via Command Completed interrupt or polling Command Completed register) before issuing the next command. However, if the Command Completed register is not set 1 second after the command is issued, software is allowed to repeat the command or to issue the next command.~~

~~If software issues a write before the controller has completed processing of the previous command and before the 1 second time limit has expired, the controller is permitted to either accept or discard the write. Regardless, such a write is considered a programming error, and could result in a discrepancy between the Slot Control register and the hot plug element state. In order to recover from such a programming error and return the controller to a consistent state, software must issue a write to the Slot Control register which conforms to the command completion rules.~~

~~Software is expected to clear the command completed event by writing a 1 to the command completed bit. However, the controller must continue to process subsequent commands normally even if the software fails to clear the command completed status bit; software must not expect further command complete interrupts unless it has cleared the command completed status bit.~~

~~IMPLEMENTATION NOTE~~ *<NOTE: THIS NOTE DELETED BY ERRATA C13>*

~~When Does a Hot-Plug Controller Set the Command Completed Register?~~

~~A command completed notification is an indication to host software that the Hot-Plug Controller has processed the previous command and is ready to receive the next command; it provides no guarantee that the action corresponding to the command is complete.~~

~~Similarly, a read to the Slot Status register following a write will always return the value from the latest write, although the corresponding command action may not be complete.~~

6.7.3.8. Port Capabilities and Slot Information Registers

Slot Implemented (PCI Express Capabilities) – ~~This bit w~~When set, this bit indicates that the Link associated with this Downstream Port is connected to a slot~~, (as oppose to being connected to an integrated device or being disabled)~~.

Physical Slot Number (Slot Capabilities) – This hardware initialized field indicates the physical slot number attached to the Port. ~~This field must be hardware initialized to a value that assigns a slot number that is globally unique within the chassis. These registers should be initialized to 0 for Ports connected to integrated devices on the system board or integrated within the same silicon as the Switch device or Root Port.~~

Hot-Plug Capable (Slot Capabilities) – ~~This bit w~~When set, this bit indicates this slot is capable of supporting ~~H~~hot-~~P~~plug.

Hot-Plug Surprise~~ Removal Capable~~ (Slot Capabilities) – ~~This bit w~~When set, this bit indicates that ~~the device~~ adapter ~~might be~~ removal~~ed~~ from the system without any prior notification is permitted for the associated form factor.

6.7.3.9. Hot-Plug Interrupt Control Register

Hot-Plug Interrupt Enable (Slot Control) – When set, t~~T~~his bit ~~when set~~ enables generation of the ~~H~~hot-~~P~~plug interrupt on enabled ~~H~~hot-~~P~~plug events.

<ed. note: The information in the following section has been comprehended above>

~~6.7.4. Hot-Plug Registers on Devices~~

~~These registers located in the Device Capabilities register are used by system software for discovery of Attention Indicator, Power Indicator, or Attention Button presence on card or module on which the device is implemented. These registers are required to be implemented on all Endpoint devices.~~

~~**Attention Button Present** – This bit when set indicates that an Attention Button is implemented on the add-in card or module.~~

~~**Attention Indicator Present** – This bit when set indicates that an Attention Indicator is implemented on the add-in card or module.~~

~~**Power Indicator Present** – This bit indicates when set indicates that a Power Indicator is implemented on the add-in card or module.~~

6.7.5. Messages

The Messages defined here allow for cards to implement indicators and buttons on the ~~card~~ adapter without having to connect signals directly to the Port. Detailed explanation of each Message is located in Chapter 2.

6.7.5.1. Messages for the Attention Indicator

This series of Messages allows the Attention Indicator to be ~~implemented~~ electrically controlled by the device on the ~~card~~ adapter as opposed to the chassis. These Messages are sent by the Downstream Port to the device and instruct the device to set its Attention Indicator to the indicated state. The following Messages are used:

~~ATTENTION_INDICATOR_ON~~Attention_Indicator_On

~~ATTENTION_INDICATOR_BLINK~~Attention_Indicator_Blink

~~ATTENTION_INDICATOR_OFF~~Attention_Indicator_Off

All Endpoint devices are required to ~~handle~~ accept without error the Attention Indicator Messages even if the device does not implement the indicator~~s~~.

6.7.5.2. Messages for Power Indicator

This series of Messages allows the Power Indicator to be ~~implemented~~ electrically controlled by the device on the ~~card~~ adapter as opposed to the chassis. These Messages are sent by the Downstream Port to the device and instruct the device to set its Power Indicator to the indicated state. The following Messages are used:

~~POWER_INDICATOR_ON~~Power_Indicator_On

~~POWER_INDICATOR_BLINK~~Power_Indicator_Blink

~~POWER_INDICATOR_OFF~~Power_Indicator_Off

All Endpoint devices are required to accept without error ~~handle~~ the Power Indicator Messages even if the device does not implement the indicator~~s~~.

6.7.5.3 Messages for Attention Button

~~ATTENTION_BUTTON_PRESSED~~Attention_Button_Pressed – This Message allows the attention button to be ~~implemented~~ electrically controlled by the device on the ~~card~~ adapter and informs the Port that the attention button has been pressed. Upon receipt of this Message the Port ~~terminates the Message and~~ sets the Attention Button Pressed bit in the ~~H~~Slot Status~~ot-Plug Event~~ register.

All down~~-~~stream Ports of Switches and Root Ports are required to accept without error ~~handle~~ the ~~ATTENTION_BUTTON_PRESSED~~ Attention_Button_Pressed Message even if the slot is not h~~H~~ot-p~~P~~lug capable.

6.7.6 Hot-Plug Slot Register Requirements

The following register fields are required for all slots that implement the ~~H~~hot-~~P~~plug capability:

- ❑ Attention Button Pressed and Attention Button Pressed Enable ~~registers~~

  Attention Button Pressed Enabled (bit 0 of Slot Control register) and Attention Button Pressed (bit 0 of Slot Status register) are required to be implemented by all ~~H~~hot-~~P~~plug capable slots to provide support for the ~~ATTENTION_BUTTON_PRESSED~~Attention_Button_Pressed Message.

- ❑ Attention Indicator Control ~~Register~~

  Attention Indicator Control ~~register~~ field (bits 7:6 of the Slot Control register) ~~is required to~~must be implemented by all ~~H~~hot-~~P~~plug capable slots to provide support for the Attention Indicator Messages.

- ❑ Power Indicator Control ~~Register~~

  Power Indicator Control ~~register~~ field (bits 9:8 of the Slot Control register) ~~is required to~~must be implemented by all ~~H~~hot-~~P~~plug capable slots to provide support for the Power Indicator Messages.

- Command Completed ~~Registers~~

  All ~~registers~~ fields associated with Command Completion (Section 6.7.3.7) must be implemented by all ~~H~~hot-~~P~~plug capable slots

- Presence Detect ~~Registers~~

  All ~~registers~~ fields associated with Presence Detect (Section 6.7.3.5) must be implemented by all ~~H~~hot-~~P~~plug capable slots

6.7.7. PCI Express Hot-Plug ~~Interrupt/Wakeup Signal Logic~~ Events

A Downstream Port with ~~H~~hot-~~P~~plug capabilities supports ~~generation of Hot-Plug interrupts on~~ the following ~~Hot-Plug~~hot-plug events:

- Slot Events:
  - Attention Button Pressed
  - Power Fault Detected
  - MRL Sensor Changed
  - Presence Detect Changed
- Command Completed Events

Each of these events has a status field, which indicates that an event has occurred but has not yet been processed by software, and an enable field, which indicates whether the event is enabled for software notification. Some events also have a capability field, which indicates whether the event type is supported on the port. The grouping of these fields by event type is listed in Section 6.7.3, and each individual field is described in Section 7.8.

6.7.7.1 Slot Events

A Downstream Port with hot-plug capabilities monitors the slot it controls for the slot events listed above. When one of these slot events is detected, the port indicates that the event has occurred by setting the status field associated with the event. At that point, the event is pending until software clears the status field.

Once a slot event is pending on a particular slot, all subsequent events of that type are ignored on that slot until the event is cleared. The port must continue to monitor the slot for all other slot event types and report them as they occur.

If enabled through the associated enable field, slot events must generate a software notification. If the event is not supported on the port as indicated by the associated capability field, software must not enable software notification for the event. The

mechanism by which this notification is reported to software is described in Section 6.7.7.3.

6.7.7.2 Command Completed Events

Since changing the state of some hot-plug elements may not happen instantaneously, PCI Express supports hot-plug commands and command completed events. All hot-plug capable ports are required to support hot-plug commands and command completed events.

Software issues a command to a hot-plug capable Downstream Port by issuing a write transaction that targets any portion of the port's Slot Control Register. A single write to the Slot Control register is considered to be a single command, even if the write affects more than one field in the Slot Control register. In response to this transaction, the port must carry out the requested actions and then set the associated status field for the command completed event. The port must process the command normally even if the status field is already set when the command is issued. Command execution must not take longer than 1 second.

Software must wait for a command to complete before issuing the next command. However, if the status field is not set after the 1 second limit on command execution, software is permitted to repeat the command or to issue the next command. If software issues a write before the port has completed processing of the previous command and before the 1 second time limit has expired, the port is permitted to either accept or discard the write. Such a write is considered a programming error, and could result in a discrepancy between the Slot Control register and the hot plug element state. To recover from such a programming error and return the controller to a consistent state, software must issue a write to the Slot Control register which conforms to the command completion rules.

If enabled through the associated enable field, the completion of a ~~commands~~ command must generate a software notification. The exception to this rule is a command that occurs as a result of a write to the Slot Control register that disables software notification of command completed events. Such a command must be processed as described above, but must not generate a software notification.

Note: The following section is modified by errata C46:

6.7.7.3 Software Notification of Hot-Plug Events

A hot-plug capable Downstream Port must support generation of an interrupt on a hot-plug event. As described in Sections 6.7.7.1 and 6.7.7.2, each type of hot-plug event has

both an enable field which enables software notification of that type of event and a status field that indicates that an event has occurred but has not yet been processed by software.

If the enable field for an event indicates that software notification of the event is disabled, the hot-plug capable Downstream Port must not generate an interrupt on that event. Hot-plug interrupt generation is also globally enabled through the Hot-Plug Interrupt Enable field in the Slot Control register. If this field indicates that hot-plug interrupts are disabled, the hot-plug capable Downstream Port must not generate an interrupt on any hot-plug event.

The Downstream Port must generate an interrupt on a hot-plug event following the rules for interrupt generation described in the rest of this document provided these disable conditions are not present and the corresponding status field for the event is not set. If the status field is set, software has not yet handled a previous event; thus, a new interrupt must not be generated.

If the port is enabled for level-triggered interrupt generation using the INTx messages, the interrupt must remain asserted as long as global interrupt generation is enabled and at least one status field for an enabled hot-plug event remains set. If the port is enabled for edge-triggered interrupt generation using MSI or MSI-X, an MSI must be sent whenever global interrupt generation is enabled and the status field for an enabled hot-plug event transitions from not set to set. The port may optionally send an MSI when there are hot-plug events that occur while interrupt generation is disabled, and interrupt generation is subsequently enabled.

If wake generation is required by the associated form factor specification, a hot-plug capable Downstream Port must support generation of a wakeup event (using the PME mechanism) on hot-plug events that occur when the system is in a sleep state or the port is in device state D1, D2, or $D3_{Hot}$.

Software enables a hot-plug event to generate a wakeup event by enabling software notification of the event as described in Section 6.7.7.1. Note that in order for software to disable interrupt generation while keeping wakeup generation enabled, the Hot-Plug Interrupt Enable bit must be cleared. For form factors that support wake generation, a wakeup event must be generated if all three of the following conditions occur:

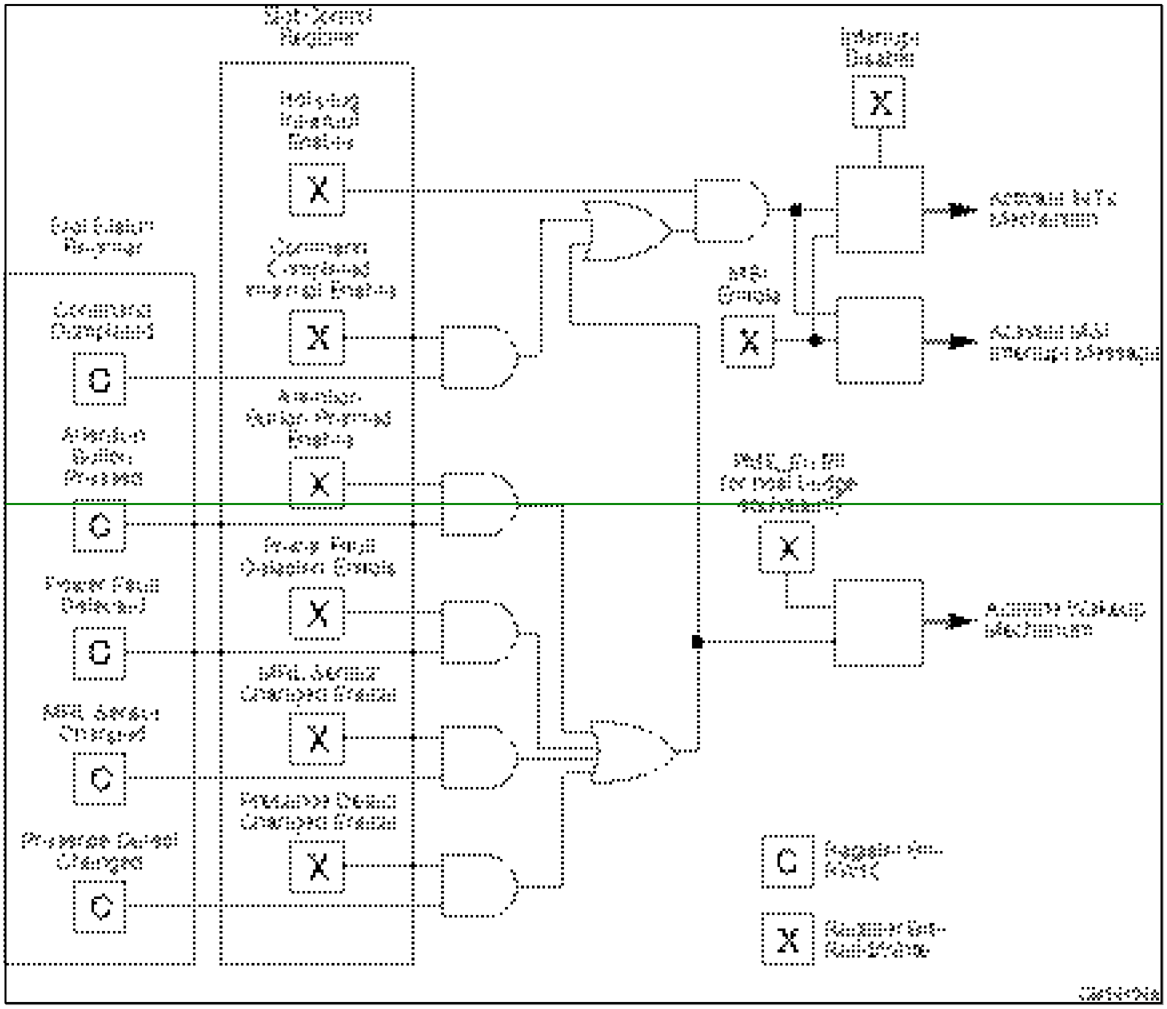
- The status register for an enabled event transitions from not set to set
- The port is in device state D1, D2, or $D3_{Hot}$, and
- The PME Enable bit in the port's Power Management Control/Status Register is set

~~When the system is in a sleep state or if the Hot-Plug capable Port is in a device state D1, D2, or $D3_{hot}$, the above Hot-Plug Controller events generate a wakeup event (using PME mechanism) instead of a Hot-Plug interrupt.~~

Note that the Hot-Plug Controller generates the wakeup on behalf of the hot-plugged device, and it is not necessary for that device to have auxiliary (or main) power.

~~A Hot-Plug capable Port also supports generation of Hot-Plug interrupt when the Hot-Plug control logic completes an issued command. However, a command completed event will not cause a wakeup event if the system is in a sleep state or if the Hot-Plug capable Port is in a device state D1, D2, or $D3_{hot}$.~~

~~Figure 6-9 shows the logical connection between the Hot-Plug event logic and the system interrupt/wakeup event generation logic.~~



~~**Figure 6-9: Hot-Plug Logic**~~

6.7.8. The Operating System Hot-Plug Method

Some systems that include ~~Hot-Plug~~hot-plug capable Root Ports and Switches that are released before ACPI-compliant operating systems with native ~~Hot-Plug~~hot-plug support are available, can use ACPI firmware for propagating ~~Hot-Plug~~hot-plug events. Firmware control of the ~~Hot-Plug~~hot-plug registers must be disabled if an operating system with native support is used. Platforms that provide ACPI firmware to propagate ~~Hot-Plug~~hot-plug events must also provide a control method to transfer control to the operating system. This method is called Operating System Hot-Plug (OSHP) and is provided for each Port that is ~~Hot-Plug~~hot-plug capable and being controlled by ACPI firmware.

Operating systems with native ~~Hot-Plug~~hot-plug support must execute the OSHP method, if present, for each ~~Hot-Plug~~hot-plug capable Port before accessing the ~~Hot-Plug~~hot-plug registers and when returning from a hibernated state. If a Port's OSHP method is executed multiple times, and the Switch to operating system control has already been achieved, the method must return successfully without doing anything. After the OSHP method is executed, the firmware must not access the Port's ~~Hot-Plug~~hot-plug registers. If any signals such as the System Interrupt or PME# have been redirected for servicing by the firmware, they must be restored appropriately for operating system control.

The following is an example of a namespace entry for an SHPC that is managed by firmware.

```
Device(PPB1){

      ...

      Method(OSHP, 0) {
            // Disable firmware access to SHPC and restore
            // the normal System Interrupt and Wakeup mechanism.
      }
      ...
}
```

IMPLEMENTATION NOTE

Controlling Hot-Plug by Using ACPI

When using ACPI to control the ~~Hot-Plug~~hot-plug events, the following should be considered:

Firmware should redirect the System Interrupt to a GPE so that ACPI can service the interrupts instead of the operating system. An appropriate _*Exx* GPE handler should be provided. When an operating system with native ~~Hot-Plug~~hot-plug support executes the OSHP method, the firmware restores the normal System Interrupt so the interrupts can be serviced by the operating system.

…

6.8. Power Budgeting Capability

With the addition of a ~~Hot-Plug~~hot-plug capability for ~~add-in cards~~adapters, the need arises ...

… The devices and/or ~~add-in cards~~ adapters are required by PCI Express to remain under the configuration power limit specified in the corresponding electromechanical specification until they have been configured and enabled by the system. The system should guarantee that power has been properly budgeted prior to enabling an ~~add-in card~~adapter.

…

6.9. Slot Power Limit Control <note – this section also modified by errata C22 (modifications non-conflicting)>

PCI Express provides a mechanism for software controlled limiting of the maximum power per slot that PCI Express ~~card/module~~adapter (associated with that slot) can consume.

…

- Partitioning of the platform, including slots for I/O expansion using ~~add-in cards/modules~~adapters
- Power delivery capabilities
- Thermal capabilities

This software is responsible for correctly programming the Slot Power Limit Value and Scale fields of the Slot Capability registers of the Downstream Ports connected to ~~I/O expansion~~ slots. After the value has been written into the register within the Downstream Port, it is conveyed to the ~~other component connected to that Port~~ adapter using the Set_Slot_Power_Limit Message (see Section 2.2.8.5). The recipient of the Message must use the value in the Message data payload to limit usage of the power for the entire ~~card/module~~adapter, unless the ~~card/module~~ adapter will never exceed the lowest value specified in the corresponding electromechanical specification. It is assumed that device driver software associated with ~~card/module~~ the adapter will be able (by reading the values of the Slot Power Limit Value and Scale fields of the Device Capability register) to configure hardware of the ~~card/module~~adapter to guarantee that the ~~card/module~~ adapter will not exceed imposed limit. In the case where the platform imposes a limit that is below minimum needed for adequate operation, the device driver will be able to communicate this discrepancy to higher level configuration software. The software is required to program one of the maximum values specified in the appropriate ~~Add-in Card/Module~~Adapter Power Dissipation Table based on the ~~card/module~~ adapter type, form factor, and capability of the system.

The following rules cover the Slot Power Limit control mechanism:

For ~~Cards/Modules~~ Adapters:

- Until and unless a Set_Slot_Power_Limit Message is received indicating a Slot Power Limit value greater than the lowest value specified in the electromechanical specification for the ~~card/module~~adapter's form factor, the ~~card/module~~ adapter must not consume more than the lowest value specified.
- A ~~card/module~~ adapter must never consume more power than what was specified in the most recently received Set_Slot_Power_Limit Message.
- Endpoint, Switch and PCI Express-PCI Bridge components that are targeted for integration on an adapter ~~card/module~~ where total consumed power is below lowest limit defined for the targeted form factor are permitted to ignore Set_Slot_Power_Limit Messages, and to return a value of 0 in the Slot Power Limit Value and Scale fields of the Device Capability register

- Such components still must be able to receive the Set_Slot_Power_Limit Message correctly but simply discard the Message

For Root Complex and Switches which source slots:

- A Downstream Port must not transmit a Set_Slot_Power_Limit Message which indicates a limit that is lower than the lowest value specified in the electromechanical specification for the slot's form factor.

IMPLEMENTATION NOTE

Example ~~Device~~Adapter Behavior Based on the Slot Power Limit Control Capability

The following power limit scenarios are examples of how an ~~device~~adapter must behave based on the Slot Power Limit control capability. The form factor limits are representations, and should not be taken as actual requirements.

Note: Form factor #1 has a maximum power requirement of 40 W and 25 W; form factor #2 has a maximum power requirement of 15 W.

Scenario 1: An ~~Device~~Adapter Consuming 12 W

- If the ~~device~~adapter is plugged into a form factor #1 40 W slot, the Slot Power Limit control mechanism is followed, and the ~~device~~adapter operates normally.
- If the ~~device~~adapter is plugged into a form factor #1 25 W slot, the Slot Power Limit control mechanism is followed, and the ~~device~~adapter operates normally.
- If the ~~device~~adapter is plugged into a form factor #2 15 W slot, the Slot Power Limit control mechanism is followed, and the ~~device~~adapter operates normally.

In all cases, since the ~~device~~adapter operates normally within all the form factors, it can ignore any of the slot power limit Messages.

Scenario 2: An ~~Device~~Adapter Consuming 18 W

- If the ~~device~~adapter is plugged into a form factor #1 40 W slot, the Slot Power Limit control mechanism is followed, and the ~~device~~adapter operates normally.
- If the ~~device~~adapter is plugged into a form factor #1 25 W slot, the Slot Power Limit control mechanism is followed, and the ~~device~~adapter operates normally.
- If the ~~device~~adapter is plugged into a form factor #2 15 W slot, the Slot Power Limit control mechanism is followed, and the ~~device~~adapter must scale down to 15 W or disable operation. An ~~device~~adapter that does not scale within any of the power limits for a given form factor will always be disabled in that form factor and should not be used.

In this case, if the ~~device~~adapter is only to be used in form factor #1, it can ignore any of the slot power limit Messages. To be useful in form factor #2, the ~~device~~adapter should be capable of scaling to the power limit of form factor #2.

Scenario 3: An ~~Device~~Adapter Consuming 30 W

- If the ~~device~~adapter is plugged into a form factor #1 40 W slot, the Slot Power Limit control mechanism is followed, and the ~~device~~adapter operates normally.
- If the ~~device~~adapter is plugged into a form factor #1 25 W slot, the Slot Power Limit control mechanism is followed, and the ~~device~~adapter must scale down to 25 W or disable operation.
- If the ~~device~~adapter is plugged into a form factor #2 15 W slot, the Slot Power Limit control mechanism is followed, and the ~~device~~adapter must scale down to 15 W or disable operation. A ~~device~~adapter that does not scale within any of the power limits for a given form factor will always be disabled in that form factor and should not be used.

In this case, since the ~~device~~adapter consumes power above the lowest power limit for a slot, the ~~device~~adapter must be capable of scaling or disabling to prevent system failures. Operation of ~~device~~adapters at power levels that exceed the capabilities of the slots in which they are plugged must be avoided.

IMPLEMENTATION NOTE

Slot Power Limit Control Registers

Typically Slot Power Limit registers within Downstream Ports of Root Complex or a Switch Device will be programmed by platform-specific software. Some implementations may use a hardware method for initializing the values of these registers and, therefore, not require software support.

Endpoint, Switch, and PCI Express-PCI Bridge components that are targeted for integration on the ~~card/module~~adapter where total consumed power is below lowest limit defined for that form factor are allowed to ignore Set_Slot_Power_Limit Messages. Note that PCI Express components that take this implementation approach may not be compatible with potential future defined form factors. Such form factors may impose lower power limits that are below the minimum required by a new ~~add-in card/module~~adapter based on the existing component.

…

7.8.3. Device Capabilities Register (Offset 04h)

The Device Capabilities register identifies PCI Express device specific capabilities. Figure 7-13 details allocation of register fields in the Device Capabilities register; Table 7-11 provides the respective bit definitions.

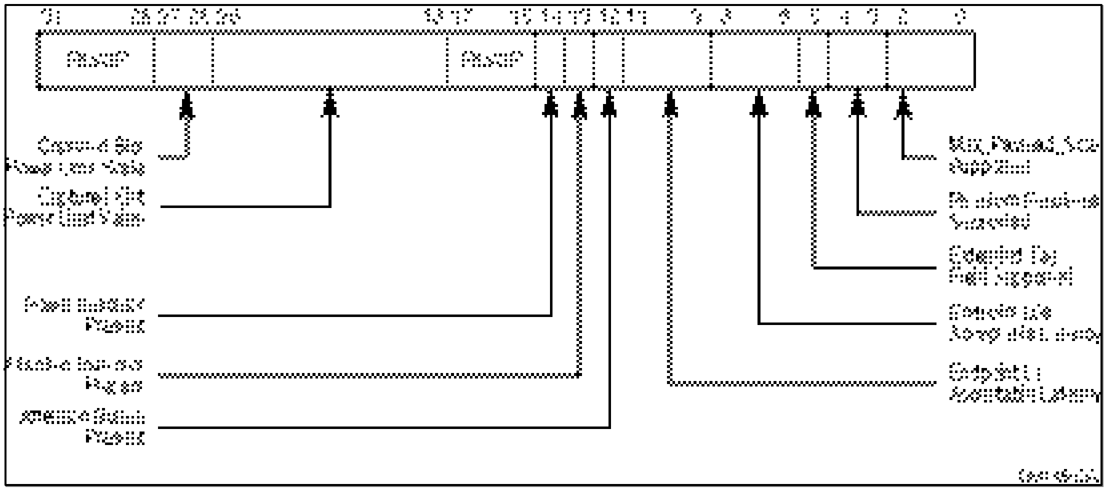


Figure 7-13: Device Capabilities Register

Table 7-11: Device Capabilities Register

Bit Location	Register Description	Attributes
…		
12	**Attention Button Present** – ~~This bit w~~When set to 1b, this bit indicates that an Attention Button is implemented on the ~~card or module~~adapter and is electrically controlled by the component on the adapter. Attention Button press events are reported using the Attention_Button_Pressed Message. This bit is valid for the following PCI Express device Types: • PCI Express Endpoint device • Legacy PCI Express Endpoint device • Upstream Port of PCI Express Switch • PCI Express-to-PCI/PCI-X Bridge	RO
13	**Attention Indicator Present** – ~~This bit, w~~When set~~,~~ to 1b, this bit indicates that an Attention Indicator is implemented on the ~~card or module~~adapter and is electrically controlled by the component on the adapter using the Attention_Indicator_On, Attention_Indicator_Blink, and Attention_Indicator_Off Messages. This bit is valid for the following PCI Express device Types: • PCI Express Endpoint device • Legacy PCI Express Endpoint device • Upstream Port of PCI Express Switch • PCI Express-to-PCI/PCI-X Bridge	RO

Bit Location	Register Description	Attributes
14	**Power Indicator Present** – ~~This bit, w~~When set~~,~~ to 1b, this bit indicates that a Power Indicator is implemented on the ~~card or module~~adapter and is electrically controlled by the component on the adapter using the Power_Indicator_On, Power_Indicator_Blink, and Power_Indicator_Off Messages. This bit is valid for the following PCI Express device Types: • PCI Express Endpoint device • Legacy PCI Express Endpoint device • Upstream Port of PCI Express Switch • PCI Express-to-PCI/PCI-X Bridge	RO
…		

…

7.8.7. Link Control Register (Offset 10h)

…

IMPLEMENTATION NOTE

Use of the Slot Clock Configuration and Common Clock Configuration Bits

In order to accurately determine the common clocking configuration of components on opposite ends of a Link that crosses a connector, there are two pieces of information that are required. The following description defines these requirements.

The first necessary piece of information is whether the Port that connects to the ~~system side of the connector~~slot uses a clock that has a common source and therefore constant phase relationship to the clock signal provided on the ~~connector~~slot. This information is provided by the system side component through a hardware initialized bit (Slot Clock Configuration) in its Link Status register. Note that some electromechanical form factor specifications may require the Port that connects to the ~~system side of the connector~~ slot use a clock that has a common source to the clock signal provided on the ~~connector~~slot.

The second necessary piece of information is whether the device on the ~~option card~~adapter uses the clock supplied on the ~~connector~~ slot or one generated locally on the ~~card~~adapter. It is the ~~card~~ adapter design and layout that will determine whether or not the Endpoint component is connected to the clock source provided by the slot~~connector~~. An Endpoint going onto this ~~card~~ adapter should have some hardware initialized method for the ~~card~~ adapter design/designer to indicate the configuration used for this particular ~~card~~ adapter design. This information is reported by the Endpoint device in bit 12 (Slot Clock Configuration) of its Link Status register. Note that some electromechanical form factor specifications may require the Port on the adapter~~that connects to the card side of the connector~~ use the clock signal provided on the connector.

…

...

7.8.9. Slot Capabilities Register (Offset 14h)

The Slot Capabilities register identifies PCI Express slot specific capabilities. Figure 7-19 details allocation of register fields in the Slot Capabilities register; Table 7-17 provides the respective bit definitions.

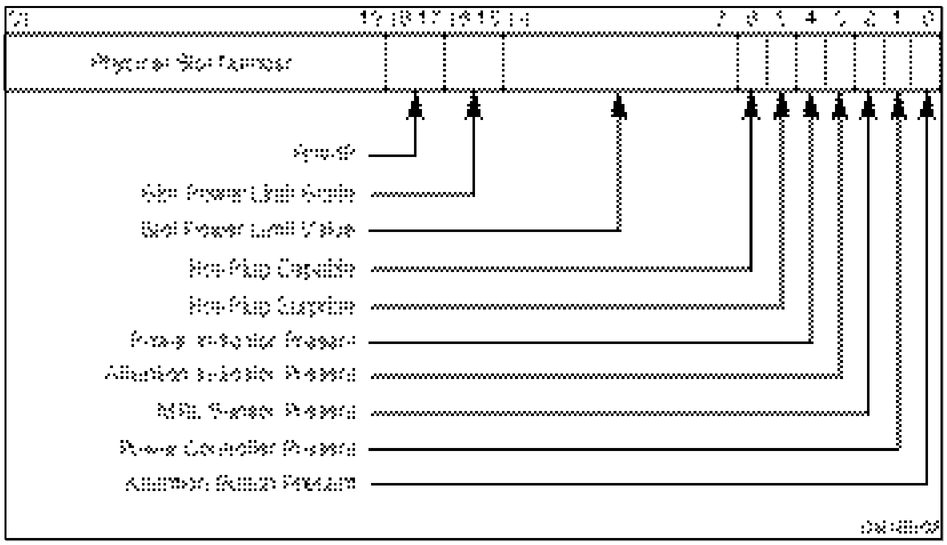


Figure 7-19: Slot Capabilities Register

Table 7-17: Slot Capabilities Register

Bit Location	Register Description	Attributes
0	**Attention Button Present** – ~~This bit w~~When set to 1b, this bit indicates that an Attention Button for this slot is ~~implemented~~ electrically controlled by ~~on~~ the chassis~~ for this slot~~.	HwInit
1	**Power Controller Present** – ~~This bit w~~When set to 1b, this bit indicates that a software programmable Power Controller is implemented for this slot/adapter (depending on form factor).	HwInit
2	**MRL Sensor Present** – ~~This bit w~~When set to 1b, this bit indicates that an MRL Sensor is implemented on the chassis for this slot.	HwInit
3	**Attention Indicator Present** – ~~This bit w~~When set to 1b, this bit indicates that an Attention Indicator is ~~implemented~~ electrically controlled by ~~on~~ the chassis~~ for this slot~~.	HwInit

Bit Location	Register Description	Attributes
4	**Power Indicator Present** – ~~This bit w~~When set to 1b, this bit indicates that a Power Indicator is ~~implemented~~ electrically controlled by ~~on~~ the chassis for this slot.	HwInit
5	**Hot-Plug Surprise** – ~~This bit w~~When set to 1b, this bit indicates that an ~~device~~ adapter present in this slot might be removed from the system without any prior notification. This is a form factor specific capability. This bit is an indication to the operating system to allow for such removal without impacting continued software operation.	HwInit
6	**Hot-Plug Capable** – ~~This bit w~~When set to 1b, this bit indicates that this slot is capable of supporting ~~Hot-Plug~~hot-plug operations.	HwInit
…		
31:19	**Physical Slot Number** – This hardware initialized field indicates the physical slot number attached to this Port. This field must be hardware initialized to a value that assigns a slot number that is ~~globally~~ unique within the chassis, regardless of the form factor associated with the slot. Th~~ese registers~~is field ~~should~~ must be initialized to 0 for Ports connected to devices that are either integrated on the system board or integrated within the same silicon as the Switch device or Root Port.	HwInit

7.8.10. Slot Control Register (Offset 18h)

The Slot Control register controls PCI Express Slot specific parameters. Figure 7-20 details allocation of register fields in the Slot Control register; Table 7-18 provides the respective bit definitions. Register fields for control parameters not implemented by the device have the RsvdP attribute.

Attention Indicator Control, Power Indicator Control, and Power Controller Control fields of the Slot Control register do not have a defined default value. It is the responsibility of the software agent (either system firmware or OS software) that runs after reset of a link to (re)initialize these fields.

In hot-plug capable Downstream Ports, a write~~s~~ to the Slot Control register must cause a hot-plug command to be generated. See Section 6.7.7.2 for details on hot-plug commands. A write to the Slot Control register in a Downstream Port~~s~~ that is not hot-plug capable must not cause a hot-plug command to be executed.

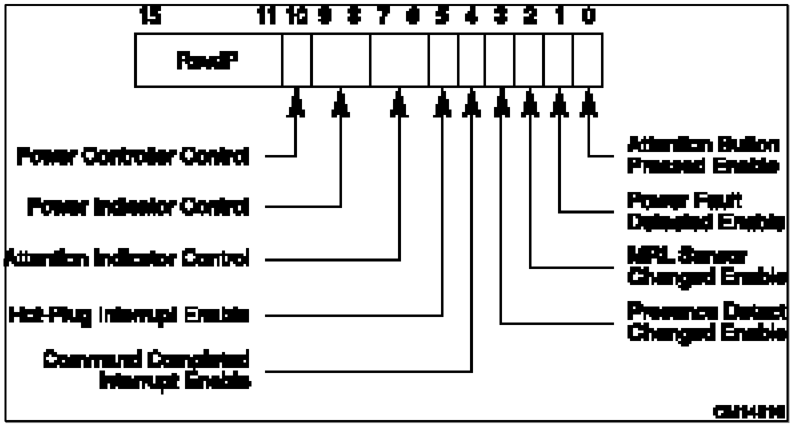


Figure 7-20: Slot Control Register

Table 7-18: Slot Control Register

Bit Location	Register Description	Attributes
0	**Attention Button Pressed Enable** – ~~This bit w~~When set to 1b, this bit enables ~~the generation of Hot-Plug~~software notification ~~interrupt or wakeup event~~ on an attention button pressed event. See Section 6.7.7. Software may set this bit even if the Attention Button Supported bit in the Slot Capabilities register is 0b, so that Attention_Button_Pressed messages can cause software notification. Default value of this field is 0b.	RW
1	**Power Fault Detected Enable** – ~~This bit w~~When set to 1b, this bit enables ~~the generation of Hot-Plug interrupt or wakeup event~~software notification on a power fault event. See Section 6.7.7. Default value of this field is 0b. If Power Fault detection is not supported, this bit is permitted to be read-only with a value of 0b.	RW
2	**MRL Sensor Changed Enable** – ~~This bit w~~When set to 1b, this bit enables ~~the generation of Hot-Plug interrupt or wakeup event~~software notification on a MRL sensor changed event. See Section 6.7.7. Default value of this field is 0b. If the MRL Sensor Present field in the Slot Capabilities register is set to 0b, this bit is permitted to be read-only with a value of 0b.	RW
3	**Presence Detect Changed Enable** – ~~This bit w~~When set to 1b, this bit enables ~~the generation of Hot-Plug interrupt or wakeup event~~software notification on a presence detect changed event. See Section 6.7.7. Default value of this field is 0b.	RW

Bit Location	Register Description	Attributes
4	**Command Completed Interrupt Enable** – ~~This bit w~~When set to 1b, this bit enables ~~the generation of Hot-Plug interrupt~~software notification when a hot-plug command is completed by the Hot-Plug Controller. Default value of this field is 0b.	RW
5	**Hot-Plug Interrupt Enable** – ~~This bit w~~When set to 1b, this bit enables generation of ~~Hot-Plug~~an interrupt on enabled ~~Hot-Plug~~hot-plug events. Default value of this field is 0b. If the Hot Plug Capable field in the Slot Capabilities register is set to 0b, this bit is permitted to be read-only with a value of 0b.	RW
7:6	**Attention Indicator Control** – If an Attention Indicator is implemented, writes to this field set the Attention Indicator to the written state. Reads of this field must reflect the value from the latest write, even if the corresponding hot-plug command is not complete, unless software issues a write without waiting for the previous command to complete in which case the read value is undefined. If the indicator is electrically controlled by chassis, the indicator is controlled directly by the downstream port through implementation specific mechanisms. If the indicator is electrically controlled by the adapter, the indicator is indirectly controlled by transmission of the appropriate Attention Indicator Message (determined by the value written). Regardless of the location of the indicator, the downstream port must transmit the appropriate Attention Indicator Message. ~~Reads to this register return the current state of the Attention Indicator; writes to this register set the Attention Indicator.~~ Defined encodings are: 00b Reserved 01b On 10b Blink 11b Off ~~Writes to this register also cause the Port to send the appropriate ATTENTION_INDICATOR_* Messages.~~	RW

<table>
<tr><th>Bit Location</th><th>Register Description</th><th>Attributes</th></tr>
<tr><td>9:8</td><td>Power Indicator Control – If a Power Indicator is implemented, writes to this field set the Power Indicator to the written state. Reads of this field must reflect the value from the latest write, even if the corresponding hot-plug command is not complete, unless software issues a write without waiting for the previous command to complete in which case the read value is undefined.

If the indicator is electrically controlled by chassis, the indicator is controlled directly by the downstream port through implementation specific mechanisms. If the indicator is electrically controlled by the adapter, the indicator is indirectly controlled by transmission of the appropriate Power Indicator Message (determined by the value written). Regardless of the location of the indicator, the downstream port must transmit the appropriate Power Indicator Message.

~~Reads to this register return the current state of the Power Indicator; writes to this register set the Power Indicator.~~

Defined encodings are:

00b Reserved
01b On
10b Blink
11b Off

~~Writes to this register also cause the Port to send the appropriate POWER_INDICATOR_* Messages.~~</td><td>RW</td></tr>
</table>

Bit Location	Register Description	Attributes
10	**Power Controller Control** – If a Power Controller is implemented, this field when written sets the power state of the slot per the defined encodings. Reads of this field must reflect the value from the latest write, even if the corresponding hot-plug command is not complete, unless software issues a write without waiting for the previous command to complete in which case the read value is undefined. Depending on the form factor, the power is turned on/off either to the slot or within the adapter. Note that in some cases the power controller may autonomously remove slot power or not respond to a power-up request based on a detected fault condition, independent of the Power Controller Control setting. ~~When read this register returns the current state of the Power applied to the slot; when written sets the power state of the slot per the defined encodings.~~ The defined encodings are: 0b Power On 1b Power Off If ~~a~~ the Power Controller Implemented field in the Slot Capabilities register is set to 0b, then writes to this field have no effect and the read value of this field is undefined.	RW

7.8.11. Slot Status Register (Offset 1Ah)

The Slot Status register provides information about PCI Express Slot specific parameters. Figure 7-21 details allocation of register fields in the Slot Status register; Table 7-19 provides the respective bit definitions. Register fields for status bits not implemented by the device have the RsvdZ attribute.

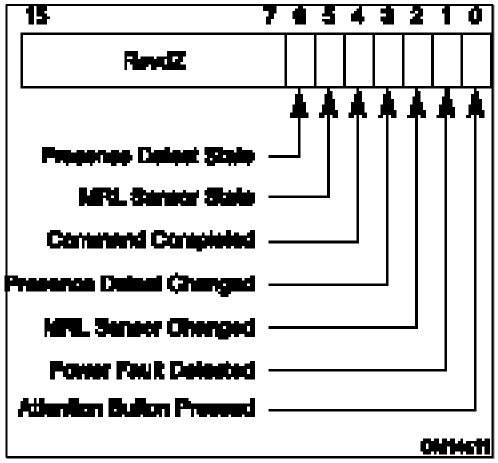


Figure 7-21: Slot Status Register

Table 7-19: Slot Status Register

Bit Location	Register Description	Attributes
0	**Attention Button Pressed** – If an Attention Button is implemented, t~~T~~his bit is set when the attention button is pressed (either as directly detected or by the receipt of an Attention_Button_Pressed Message). If an Attention Button is not supported and the form factor does not support the Attention_Button_Pressed Message, this bit must not be set.	RW1C
1	**Power Fault Detected** – If a Power Controller that supports power fault detection is implemented, t~~T~~his bit is set when the Power Controller detects a power fault at this slot.– Note that, depending on hardware capability, it is possible that a power fault can be detected at any time, independent of the Power Controller Control setting or the occupancy of the slot. If power fault detection is not supported, this bit must not be set.	RW1C
2	**MRL Sensor Changed** – If an MRL sensor is implemented, t~~T~~his bit is set when a MRL Sensor state change is detected. If an MRL sensor is not implemented, this bit must not be set.	RW1C
3	**Presence Detect Changed** – This bit is set when ~~a~~ the value reported in Presence Detect State is changed~~ is detected~~.	RW1C
4	**Command Completed** – This bit is set when a hot-plug command has completed and the Hot-Plug Controller ~~completes an issued command~~is ready to accept a subsequent command. The Command Completed status bit is set as an indication to host software that the Hot-Plug Controller has processed the previous command and is ready to receive the next command; it provides no guarantee that the action corresponding to the command is complete.	RW1C

Bit Location	Register Description	Attributes
5	**MRL Sensor State** – This register reports the status of the MRL sensor if ~~it is~~ implemented. Defined encodings are: 0b MRL Closed 1b MRL Open	RO
6	**Presence Detect State** – ~~This bit indicates the presence of a card in the slot.~~ This bit indicates the presence of an adapter in the slot, reflected by the logical "OR" of the Physical Layer in-band presence detect mechanism and, if present, any out-of-band presence detect mechanism defined for the slot's corresponding form factor. ~~The bit reflects the Presence Detect status determined via an in-band mechanism or via the Present Detect pins as defined in the *PCI Express Card Electromechanical Specification*.~~ Note that the in-band presence detect mechanism requires that power be applied to an adapter for its presence to be detected. Consequently, form factors that require a power controller for hot-plug must implement a physical pin presence detect mechanism. Defined encodings are: 0b Slot Empty 1b Card Present in slot ~~This register is required to be implemented on all Switch devices and Root Ports. The presence detect field for Switch devices or Root Ports~~ This register must be implemented on all Downstream Ports that implement slots. For Downstream Ports not connected to slots (where the **Slot Implemented** bit of the PCI Express Capabilities Register is 0b) ~~should~~ this bit must return 1b. ~~be hardwired to 1. This register is required if a slot is implemented.~~	RO

IMPLEMENTATION NOTE

No Slot Power Controller

For slots that do not implement a power controller, software must ensure that system power planes are enabled to provide power to slots prior to reading presence detect state.

C22. Slot Power Limit Operation

PWG Approved 4 Mar 2004 -- Release Date: 12 Jul 2004

In 2.2.8.5. Slot Power Limit Support:

... This Message ~~is~~must be sent automatically by the Downstream Port (of a Root Complex or a Switch) when one of the following events occurs:

- On a Configuration Write to the Slot Capabilities register (see Section 7.8.9) when the Data Link Layer reports DL_Up status.
- Anytime when a Link transitions from a non-DL_Up status to a DL_Up status (see Section 2.9.2). This Transmission is optional if the Slot Capabilities Register has not yet been initialized.

The component on the other side of the Link (Endpoint, Switch, or Bridge) that receives Set_Slot_Power_Limit Message must copy the values in the data payload into the Device Capabilities register associated with the component's Upstream Port. PCI Express components that are targeted exclusively for integration on the system planar (e.g., system board) as well as components that are targeted for integration on a card/module where power consumption of the entire card/module is below the lowest power limit specified for the card/module form factor (as defined in the corresponding ~~electromechanical~~form factor specification) are permitted to hardwire the value 0b in the Slot Power Limit Scale and Slot Power Limit Value fields of the Device Capabilities register, and are not required to copy the Set_Slot_Power limit payload into that register.

For more details on Power Limit control mechanism see Section 6.9.

In 2.9.2. Transaction Layer Behavior in DL_Up Status (Note: this text also affected by C17 & C21:

DL_Up status indicates that a connection has been established with another component on the associated Link. This section specifies the Transaction Layer's behavior when the Data Link Layer reports entry to the DL_Up status to the Transaction Layer, indicating that the Link is operational. These behaviors relate to Slot Power Limit support.

For a Downstream Port on a Root Complex or a Switch:

- When transitioning from a non-DL_Up Status to a DL_Up Status, the Port must initiate the transmission of a Set_Slot_Power_Limit Message to the other component on the Link to convey the value programmed in the Slot Power Limit Scale and Value fields of the Slot Capabilities register. This Transmission is optional if the Slot Capabilities Register has not yet been initialized.

In 6.9 Slot Power Limit Control (Note: this text also affected by C21):

... The recipient of the Message must use the value in the Message data payload to limit usage of the power for the entire card/module, unless the card/module will never exceed the lowest value specified in the corresponding ~~electromechanical~~form factor specification. It is ~~assumed~~required that device driver software associated with card/module ~~will~~ be able (by reading the values of the Slot Power Limit Value and Scale fields of the Device Capability register) to configure hardware of the card/module to guarantee that the card/module will not exceed the imposed limit. In the case where the platform imposes a limit that is

below the minimum needed for adequate operation, the device driver will be able to communicate this discrepancy to higher level configuration software. ~~The~~ Configuration software is required to ~~program~~ set the Slot Power Limit to one of the maximum values specified ~~in~~ for the ~~appropriate~~ corresponding form factor ~~Add-in Card/Module Power Dissipation Table~~ based on the ~~card/module type, form factor, and~~ capability of the ~~system~~ platform.

The following rules cover the Slot Power Limit control mechanism:

For Cards/Modules:

- Until and unless a Set_Slot_Power_Limit Message is received indicating a Slot Power Limit value greater than the lowest value specified in the ~~electromechanical~~ form factor specification for the card/module's form factor, the card/module must not consume more than the lowest value specified.
- A card/module must never consume more power than what was specified in the most recently received Set_Slot_Power_Limit Message, or the minimum value specified in the corresponding form factor specification, whichever is higher.
- Endpoint, Switch and PCI Express-PCI Bridge components that are targeted for integration on a card/module where total consumed power is below the lowest limit defined for the targeted form factor are permitted to ignore Set_Slot_Power_Limit Messages, and to return a value of 0 in the Slot Power Limit Value and Scale fields of the Device Capability register
    - *[note bullet demotion]* Such components still must be able to receive the Set_Slot_Power_Limit Message ~~correctly~~ without error, but simply discard the Message value

For Root Complex and Switches which source slots:

- ~~A Downstream Port must not transmit~~ Configuration software must not program a Set_Slot_Power_Limit value ~~Message~~ which indicates a limit that is lower than the lowest value specified in the ~~electromechanical~~ form factor specification for the slot's form factor.

C23. AER, Completion Timeout and Switches

PWG Approved 4 Mar 2004 -- Release Date: 12 Jul 2004

In Section 7.10.2. Uncorrectable Error Status Register (Offset 04h):

Table 7-24: Uncorrectable Error Status Register

Bit Location	Register Description	Attributes	Default Value
…			

Bit Location	Register Description	Attributes	Default Value
14	**Completion Timeout Status**[Footnote: For Switch Ports, required if the Switch Port issues Non-Posted Requests on its own behalf (vs. only forwarding such Requests generated by other devices). If the Switch Port does not issue such Requests, then the Completion Timeout mechanism is not applicable and this bit must be hardwired to 0b.]	RW1CS	0
…			

In Section 7.10.3. Uncorrectable Error Mask Register (Offset 08h):

Table 7-25: Uncorrectable Error Mask Register

Bit Location	Register Description	Attributes	Default Value
…			
14	**Completion Timeout Mask**[Footnote: For Switch Ports, required if the Switch Port issues Non-Posted Requests on its own behalf (vs. only forwarding such Requests generated by other devices). If the Switch Port does not issue such Requests, then the Completion Timeout mechanism is not applicable and this bit must be hardwired to 0b.]	RWS	0
…			

In Section 7.10.4. Uncorrectable Error Severity Register (Offset 0Ch):

Table 7-26: Uncorrectable Error Severity Register

Bit Location	Register Description	Attributes	Default Value
…			
14	**Completion Timeout Error Severity**[Footnote: For Switch Ports, required if the Switch Port issues Non-Posted Requests on its own behalf (vs. only forwarding such Requests generated by other devices). If the Switch Port does not issue such Requests, then the Completion Timeout mechanism is not applicable and this bit must be hardwired to 0b.]	RWS	0
…			

C24. Compliance Pattern

EWG Approved -- Release Date: 12 Jul 2004

In section 4.2.8 Change:

This delay sequence on every eighth Lane is then

...

...												
Lane 4	K28.5-	D21.5	K28.5+	D10.2	K28.5-	D21.5	K28.5+	D10.2	K28.5-	D21.5	K28.5+	D10.2
...												

C25. Link Down, Hot Rest and Fundamental Reset

PWG Approved 23 Feb 2004 -- Release Date: 12 Jul 2004

In Section 6.6, PCI Express Reset – Rules:

...

- On exit from any type of reset (cold, warm, or hot), all Port registers and state machines must be set to their initialization values as specified in this document, except for sticky registers (see Section 7.4).
  - Note that, from a device point of view, any type of reset (cold, warm, hot or DL_Down) has the same effect at the Transaction Layer and above as would RST# assertion and de-assertion in conventional PCI.

C26. Link Behavior for Multi-Function Devices

PWG Approved 4 Mar 2004 -- Release Date: 12 Jul 2004

In section 7.1 insert the following at the end of the last paragraph.

A PCI Express Endpoint is mapped into configuration space as a single logical device with one or more logical functions. Unless otherwise specified, requirements in the configuration space definition for a device apply to single function devices as well as to each function individually of a multi-function device.

C27. Withdrawn

C28. Secondary Bus Reset Timing Requirements

PWG Approved 18 Mar 2004 -- Release Date: 12 Jul 2004

Edit as shown:

7.5.3.5. Bridge Control Register (Offset 3Eh)

Table 7-6 establishes the mapping between PCI 2.3 and PCI Express for the PCI 2.3 configuration space Bridge Control register.

Table 7-6: Bridge Control Register

Bit Location	Register Description	Attributes
…	**…**	…
6	**Secondary Bus Reset** – Setting this bit triggers a hot reset on the corresponding PCI Express Port. Software must ensure a minimum reset duration (T_{rst}) as defined in the PCI 2.3 Specification. Software and systems must honor first-access-following-reset timing requirements defined in Section 6.6. Port configuration registers must not be affected, except as required to update Port status. Default value of this field is 0.	RW
…	**…**	…

C29. Max_Payload_Size and Multi-Function Devices

PWG Approved 8 Apr 2004 -- Release Date: 12 Jul 2004

2.2.2. TLPs with Data Payloads - Rules

...

- The Transmitter of a TLP with a data payload must not allow the data payload length to exceed the length specified by the value in the Max_Payload_Size field of the Transmitter's Device Control register (see Section 7.8.4).
  - For an Upstream Port associated with a multi-function device whose Max_Payload_Size settings are identical across all functions, a transmitted TLP's data payload must not exceed the common Max_Payload_Size setting.
  - For an Upstream Port associated with a multi-function device whose Max_Payload_Size settings are not identical across all functions, a transmitted TLP's

data payload must not exceed a Max_Payload_Size setting whose determination is implementation specific.

- ? Transmitter implementations are encouraged to use the Max_Payload_Size setting from the function that generated the transaction, or else the smallest Max_Payload_Size setting across all functions.
- ? Software should not set the Max_Payload_Size in different functions to different values unless software is aware of the specific implementation.

~~?~~ o Note: Max_Payload_Size applies only to TLPs with data payloads; Memory Read Requests are not restricted in length by Max_Payload_Size. The size of the Memory Read Request is controlled by the Length field

- ❑ The data payload of a Received TLP must not exceed the length specified by the value in the Max_Payload_Size field of the Receiver's Device Control register (see Section 7.8.4).

~~?~~ o Receivers must check for violations of this rule. If a Receiver determines that a TLP violates this rule, the TLP is a Malformed TLP

- ? This is a reported error associated with the Receiving Port (see Section 6.2)

o For an Upstream Port associated with a multi-function device whose Max_Payload_Size settings are identical across all functions, the Receiver is required to check the TLP's data payload size against the common Max_Payload_Size setting.

o For an Upstream Port associated with a multi-function device whose Max_Payload_Size settings are not identical across all functions, the Receiver is required to check the TLP's data payload against a Max_Payload_Size setting whose determination is implementation specific.

- ? Receiver implementations are encouraged to use the Max_Payload_Size setting from the function targeted by the transaction, or else the largest Max_Payload_Size setting across all functions.
- ? Software should not set the Max_Payload_Size in different functions to different values unless software is aware of the specific implementation.

...

2.3.1.1. Data Return for Read Requests

...

- ❑ Completions must not include more data than permitted by the Max_Payload_Size parameter.

~~?~~ o Receivers must check for violations of this rule~~ — TLPs in violation are Malformed TLPs~~. See Section 2.2.2.

~~? This is a reported error associated with the Receiving Port (see Section 6.2)~~

Note: This is simply a ~~special~~ specific case of the rules which apply to all TLPs with data payloads

...

2.6.1. Flow Control Rules

…

Table 2-27: Minimum Initial Flow Control Advertisements

Credit Type	**Minimum Advertisement**
…	…
PD	Largest possible setting of the Max_Payload_Size for the component divided by FC Unit Size. For a multi-function device, this includes all functions in the device. Example: If the largest Max_Payload_Size value supported is 1024 bytes, the smallest permitted initial credit value would be 040h.
…	…

…

2.6.1.2. FC Information Tracked by Receiver

…

- For non-inifinite PD and CPLD types, when the number of available credits is less than Max_Payload_Size, an UpdateFC FCP must be scheduled for Transmission each time one or more units of that type are made available by TLPs processed
  - For a multi-function device whose Max_Payload_Size settings are identical across all functions, the common Max_Payload_Size setting or larger must be used.
  - For a multi-function device whose Max_Payload_Size settings are not identical across all functions, the selected Max_Payload_Size setting is implementation specific, but it's recommended to use the largest Max_Payload_Size setting across all functions.

…

IMPLEMENTATION NOTE

Flow Control Update Frequency

…

The values are calculated as a function of the largest TLP payload size and Link width. The values are measured at the Port of the TLP Receiver, starting with the time the last Symbol of a TLP in received to the first Symbol of the UpdateFC DLLP being transmitted. The values are calculated using the formula:

$$\frac{(Max\_Payload\_Size + TLPOverhead) * UpdateFactor}{LinkWidth} + InternalDelay$$

where:

Max_Payload_Size	The value in the Max_Payload_Size field of the Device Control register. For a multi-function device, it is recommended that the smallest Max_Payload_Size setting across all functions be used.

…

3.5.2.1. LCRC and Sequence Number Rules (TLP Transmitter)

…

The values are measured at the Port of the TLP Transmitter, from last Symbol of TLP to First Symbol of TLP retransmission. The values are calculated using the formula (note – this is simply three times the Ack Latency value – see Section 3.5.3.1):

$$\left(\frac{(Max\_Payload\_Size + TLPOverhead) * AckFactor}{LinkWidth} + InternalDelay\right) * 3 + Rx\_L0s\_Adjustment$$

where

Max_Payload_Size	is the value in the Max_Payload_Size field of the Device Control register. For a multi-function device whose Max_Payload_Size settings are identical across all functions, the common Max_Payload_Size setting must be used. For a multi-function device whose Max_Payload_Size settings are not identical across all functions, the selected Max_Payload_Size setting is implementation specific, but it's recommended to use the largest Max_Payload_Size setting across all functions.

…

3.5.3.1. LCRC and Sequence Number Rules (TLP Receiver)

…

Table 3-5 defines the threshold values for the AckNak_LATENCY_TIMER timer, which for any specific case is called the Ack Latency. The values are specified according to the largest TLP payload size and Link width. The values are measured at the Port of the TLP Receiver, starting with the time the last Symbol of a TLP is received to the first Symbol of the Ack/Nak DLLP being transmitted. The values are calculated using the formula:

$$\frac{(Max\_Payload\_Size + TLPOverhead) * AckFactor}{LinkWidth} + InternalDelay + Tx\_L0s\_Adjustment$$

where

Max_Payload_Size	is the value in the Max_Payload_Size field of the Device Control register. For a multi-function device whose Max_Payload_Size settings are identical across all functions, the common Max_Payload_Size setting must be used. For a multi-function device whose Max_Payload_Size settings are not identical across all functions, the selected Max_Payload_Size setting is implementation specific, but it's recommended to use the smallest Max_Payload_Size setting across all functions.

…

7.8.3. Device Capabilities Register (Offset 04h)

The Device Capabilities register identifies PCI Express device specific capabilities. Figure 7-13 details allocation of register fields in the Device Capabilities register; Table 7-11 provides the respective bit definitions.

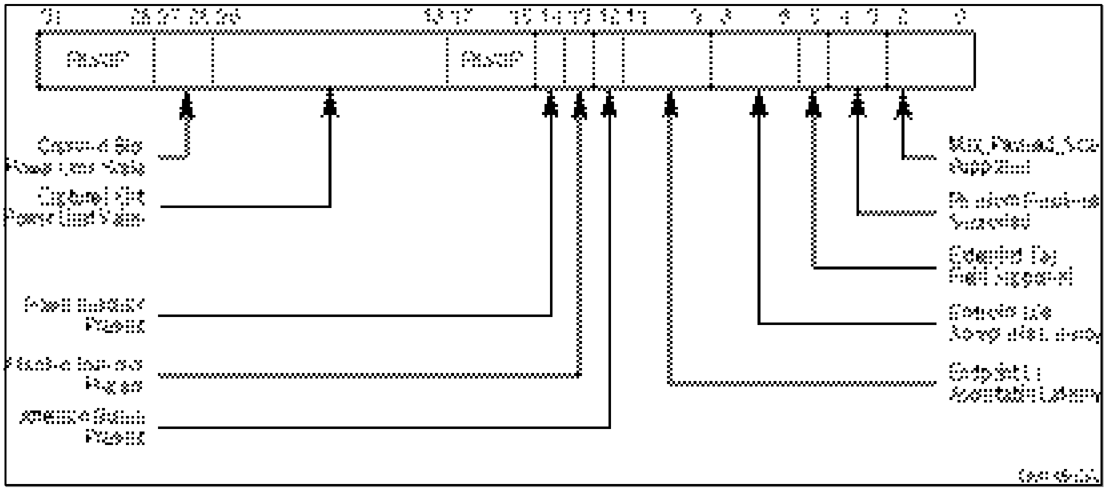


Figure 7-13: Device Capabilities Register

Table 7-11: Device Capabilities Register

Bit Location	Register Description	Attributes
2:0	**Max_Payload_Size Supported** – This field indicates the maximum payload size that the device/function can support for TLPs. Defined encodings are: 000b 128 bytes max payload size 001b 256 bytes max payload size 010b 512 bytes max payload size 011b 1024 bytes max payload size 100b 2048 bytes max payload size 101b 4096 bytes max payload size 110b Reserved 111b Reserved The functions of a multi-function device are permitted to report different values for this field.	RO
…		

...

7.8.4. Device Control Register (Offset 08h)

The Device Control register controls PCI Express device specific parameters. Figure 7-14 details allocation of register fields in the Device Control register; Table 7-12 provides the respective bit definitions.

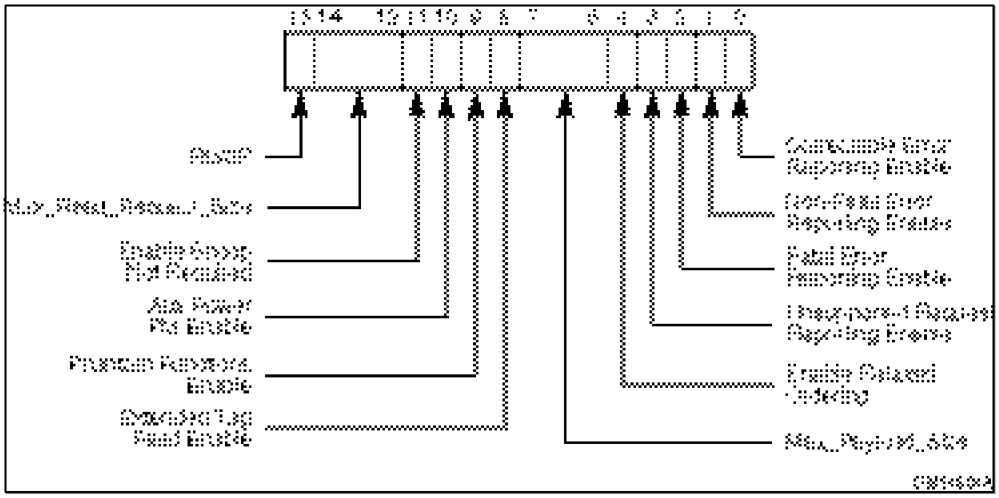


Figure 7-14: Device Control Register

Table 7-12: Device Control Register

Bit Location	Register Description	Attributes
…		

Bit Location	Register Description	Attributes
7:5	**Max_Payload_Size** – This field sets maximum TLP payload size for the device/function. As a Receiver, the device must handle TLPs as large as the set value; as Transmitter, the device must not generate TLPs exceeding the set value. Permissible values that can be programmed are indicated by the Max_Payload_Size Supported in the Device Capabilities register (refer to Section 7.8.3). Defined encodings for this field are: 000b 128 bytes max payload size 001b 256 bytes max payload size 010b 512 bytes max payload size 011b 1024 bytes max payload size 100b 2048 bytes max payload size 101b 4096 bytes max payload size 110b Reserved 111b Reserved Default value of this field is 000b. System software is not required to program the same value for this field for all the functions of a multi-function device. See Section 2.2.2 for important guidance.	RW
…		

...

C30. Errata to the Integrated Devices Event Collector ECN

SWWG Approved 21 Apr 2004 -- Release Date: 12 Jul 2004

In Section 7.5.1.1 change the following text:

7.5.1.1. Command Register (Offset 04h)

Table 7-3 establishes the mapping between PCI 2.3 and PCI Express for PCI 2.3 configuration space Command register.

Table 7-3: Command Register

Bit Location	Register Description	Attributes

…		
6	**Parity Error Enable** – See Section 7.5.1.7. A Root Complex Integrated Endpoint that is not associated with a Root Complex Event Collector is permitted to hardwire this bit to 0b. Default value of this field is 0.	RW
…		
8	**SERR Enable** – See Section 7.5.1.7. This bit, when set, enables reporting of Non-fatal and Fatal errors detected by the device to the Root Complex. Note that errors are reported if enabled either through this bit or through the PCI-Express specific bits in the Device Control register (see Section 7.8.4). A Root Complex Integrated Endpoint that is not associated with a Root Complex Event Collector is permitted to hardwire this bit to 0b. Default value of this field is 0.	RW
…		

In Section 7.8.4 change the following text:

Table 7-12: Device Control Register

Bit Location	Register Description	Attributes
0	**Correctable Error Reporting Enable** – This bit controls reporting of correctable errors. Refer to Section 6.2 for further details. For a multi-function device, this bit controls error reporting for each function from point-of-view of the respective function. For a Root Port, the reporting of correctable errors is internal to the root. No external ERR_COR Message is generated. A Root Complex Integrated Endpoint that is not associated with a Root Complex Event Collector is permitted to hardwire this bit to 0b. Default value of this field is 0.	RW
1	**Non-Fatal Error Reporting Enable** – This bit controls reporting of Non-fatal errors. Refer to Section 6.2 for further details. For a multi-function device, this bit controls error reporting for each function from point-of-view of the respective function.	RW

	For a Root Port, the reporting of Non-fatal errors is internal to the root. No external ERR_NONFATAL Message is generated. A Root Complex Integrated Endpoint that is not associated with a Root Complex Event Collector is permitted to hardwire this bit to 0b. Default value of this field is 0.	
2	**Fatal Error Reporting Enable** – This bit controls reporting of Fatal errors. Refer to Section 6.2 for further details. For a multifunction device, this bit controls error reporting for each function from point-of-view of the respective function. For a Root Port, the reporting of Fatal errors is internal to the root. No external ERR_FATAL Message is generated. A Root Complex Integrated Endpoint that is not associated with a Root Complex Event Collector is permitted to hardwire this bit to 0b. Default value of this field is 0.	RW
3	**Unsupported Request Reporting Enable** – This bit enables reporting of Unsupported Requests when set. Refer to Section 6.2 for further details. For a multi-function device, this bit controls error reporting for each function from point-of-view of the respective function. Note that the reporting of error Messages (ERR_COR, ERR_NONFATAL, ERR_FATAL) received by Root Port is controlled exclusively by Root Control register described in Section 7.8.12. A Root Complex Integrated Endpoint that is not associated with a Root Complex Event Collector is permitted to hardwire this bit to 0b. Default value of this field is 0.	RW
…		

C31. Errata to the Root Complex Topology Discovery ECN

SWWG Approved 9 May 2004 -- Release Date: 12 Jul 2004

The following changes apply to the Root Complex Topology Discovery ECN to the PCI Express Base Specification, Rev 1.0a. These changes synchronize the text in the RC TD ECN with equivalent language in the PCI Express Base spec 1.0a as modified by errata items C8 and C19 in this document.

In Section 7.14.2, modify Table 7-55 as follows:

Table 7-55: Root Complex Link Capabilities Register

Bit Location	Register Description	Attributes

Bit Location	Register Description	Attributes
3:0	**Maximum Link Speed** – This field indicates the maximum Link speed of the given Link. Defined encodings are: 0001b 2.5 Gb/s Link All other encodings are reserved. A Root Complex that does not support this feature must report~~s~~ ~~a~~ 0000b in this field.	RO
9:4	**Maximum Link Width** – This field indicates the maximum width of the given Link. Defined encodings are: 000001b X1 000010b X2 000100b X4 001000b X8 001100b X12 010000b X16 100000b X32 All other encodings are reserved. A Root Complex that does not support this feature must report~~s~~ ~~a~~ 000000b in this field.	RO
11:10	**Active State ~~Link~~ Power Management (ASPM) Support** – This field indicates the level of ~~active state power management~~ASPM supported on the given Link. Defined encodings are: 00b No A~~ctive~~ S~~tate~~ PM Support 01b L0s Entry Supported 10b L1 Entry Supported 11b L0s and L1 Supported	RO

Bit Location	Register Description	Attributes
14:12	**L0s Exit Latency** – This field indicates the L0s exit latency for the given Link. The value reported indicates the length of time this Port requires to complete transition from L0s to L0. Defined encodings are: 000b Less than 64 ns 001b 64 ns to less than 128 ns 010b 128 ns to less than 256 ns 011b 256 ns to less than 512 ns 100b 512 ns to less than 1 µs 101b 1µs to less than 2 µs 110b 2 µs-4 µs 111b ~~L0s transition not supported~~More than 4 µs	RO
17:15	**L1 Exit Latency** – This field indicates the L1 exit latency for the given Link. The value reported indicates the length of time this Port requires to complete transition from L1 to L0. Defined encodings are: 000b Less than 1 µs 001b 1 µs to less than 2 µs 010b 2 µs to less than 4 µs 011b 4 µs to less than 8 µs 100b 8 µs to less than 16 µs 101b 16 µs to less than 32 µs 110b 32 µs~~ to less than ~~-64 µs 111b ~~L1 transition not supported~~More than 64 µs	RO

In Section 7.14.3, modify Table 7-56 as follows:

Table 7-56: Root Complex Link Control Register

Bit Location	Register Description	Attributes
1: 0	**Active State ~~Link~~ Power Management (ASPM) Control** – This field controls the level of ~~active state~~ ASPM supported on the given Link. Defined encodings are: 00b Disabled 01b L0s Entry Enabled 10b L1 Entry Enabled 11b L0s and L1 Entry Enabled Note: "L0s Entry Enabled" indicates the Transmitter entering L0s is supported. The Receiver must be capable of entering L0s even when the field is disabled (00b). Default value of this field is implementation specific. ASPM L1 must be enabled by software in the upstream component on a link prior to enabling ASPM L1 in the downstream component on that link. When disabling ASPM L1, software must disable ASPM L1 in the downstream component on a link prior to disabling ASPM L1 in the upstream component on that link. ASPM L1 must only be enabled on the downstream component if both components on a link support ASPM L1. A Root Complex that does not support this feature for the given internal link must hardwire~~s~~ this field to 00b.	RW
7	**Extended Sync** – This bit when set forces ~~extended~~ the transmission of ~~FTS~~ additional ordered sets ~~in FTS and extra TS2 at exit from L1 prior to entering L0~~when exiting the L0s state (see Section 4.2.4.3) and when in the Recovery state (see Section 4.2.6.4.1). This mode provides external devices (e.g., logic analyzers) monitoring the ~~l~~Link time to achieve bit and ~~s~~Symbol lock before the ~~l~~Link enters the L0 state and resumes communication. Default value for this bit is 0b. A Root Complex that does not support this feature for the given internal link must hardwire~~s~~ this field to 0b.	RW

In Section 7.14.4, modify Table 7-57 as follows:

Table 7-57: Root Complex Link Status Register

Bit Location	Register Description	Attributes
3:0	**Link Speed** – This field indicates the negotiated Link speed of the given Link. Defined encodings are: 0001b 2.5 Gb/s Link All other encodings are reserved. The value in this field is undefined when the Link is not up. A Root Complex that does not support this feature must report~~s~~ ~~a~~ 0000b in this field.	RO
9:4	**Negotiated Link Width** – This field indicates the negotiated width of the given Link. Defined encodings are: 000001b X1 000010b X2 000100b X4 001000b X8 001100b X12 010000b X16 100000b X32 All other encodings are reserved. The value in this field is undefined when the Link is not up. A Root Complex that does not support this feature must report~~s~~ ~~a~~ 000000b in this field.	RO

C32. PCI 2.3 References

PWG Approved 20 May 2004 -- Release Date: 12 Jul 2004

Throughout the entire PCI Express Base Specification 1.0a, globally replace all references to "PCI 2.3" with "PCI 3.0".

In addition, edit the following sections in the MSI-X ECN for PCI Express:

Terms and Acronyms, p. 22, change text as shown:

Message Signaled Interrupt (MSI/MSI-X) Two similar but separate mechanisms that enable a device to request service by writing a system-specified DWORD of data to a system-specified address using a Memory Write Request. ~~MSI is~~

~~defined in PCI 2.3. MSI-X is a separate extension mechanism defined in the MSI-X ECN for PCI 2.3.~~ Compared to MSI, MSI-X supports a larger maximum number of vectors and independent message address and data for each vector.

6.1. Interrupt and PME Support, p. 259, change text as shown:

…

In addition to PCI INTx compatible interrupt emulation, PCI Express requires support of MSI or MSI-X or both. ~~The PCI Express MSI mechanism is compatible with the MSI capability defined in the PCI 2.3 Specification. The PCI Express MSI-X mechanism is compatible with the MSI-X capability defined in the MSI-X ECN for the PCI 2.3 Specification.~~The PCI Express MSI and MSI-X mechanisms are compatible with those defined in the PCI 3.0 Specification.

C33. Device-Specific Registers in Config Space

SWWG Approved 19 May 2004 -- Release Date: 12 Jul 2004

Add the following Implementation Note to 7.2.2.2. PCI Express Device Requirements:

IMPLEMENTATION NOTE

Device-Specific Registers in Configuration Space

It is strongly recommended that PCI Express devices place no registers in Configuration Space other than those in Headers or Capability structures architected by applicable PCI specifications.

Device-specific registers that have legitimate reasons to be placed in Configuration Space (e.g., they need to be accessible before Memory Space is allocated) should be placed in a Vendor-Specific Capability Structure (in PCI Compatible Configuration Space) or a Vendor-Specific Extended Capability Structure (in PCI Express Extended Configuration Space).

Device-specific registers accessed in the run-time environment by drivers should be placed in Memory Space that is allocated by one or more Base Address Registers. Even though PCI Compatible or PCI Express Extended Configuration Space may have adequate room for run-time device-specific registers, placing them there is highly discouraged for the following reasons:

- Not all Operating Systems permit drivers to access Configuration Space directly.
- Some platforms provide access to Configuration Space only via firmware calls, which typically have substantially lower performance compared to mechanisms for accessing Memory Space.
- Even on platforms that provide direct accesss to a memory-mapped PCI Express Enhanced Configuration Mechanism, performance for accessing Configuration Space will typically be significantly lower than for accessing Memory Space since:
    - Configuration Reads and Writes must usually be DWORD or smaller in size,
    - Configuration Writes are usually not posted by the platform, and
    - Some platforms support only one outstanding Configuration Write at a time.

C34. ASPM Default Values

SWWG Approved 9 May 2004 -- Release Date: 12 Jul 2004

In Section 5.2, modify the third bullet as follows:

- L1 – Higher latency, lower power "standby" state.

  L1 support is required for PCI-PM compatible power management. L1 is optional for ASPM unless specifically required by a particular form factor.

In Section 5.2, change Note 2 of Table 5-1 as follows:

2. …
3. L1 entry may be requested within ASPM protocol, however its support is optional unless specifically required by a particular form factor.

In Section 5.3.2, change Note 1 of Table 5-2 as follows:

~~11.~~1. All PCI Express components are required to support ASPM with L0s entry during idle at a minimum. The use of L1 within D0 is optional unless specifically required by a particular form factor.

In Section 5.4.1, change the 4th paragraph as follows:

The L1 Link state is optimized for maximum power savings at a cost of longer entry and exit latencies. L1 reduces Link power beyond the L0s state for cases where very low power is required and longer transition times are acceptable. ASPM support for the L1 Link state is optional unless specifically required by a particular form factor.

In Section 5.4.1.3, change the 2nd paragraph as follows:

Each PCI Express component reports its level of support for ASPM in the ASPM Support configuration field below. All PCI Express components must support transition to the L0s Link state. Support for transition to the L1 Link state while in $D0_{active}$ state is optional unless specifically required by a particular form factor.

In Section 7.8.7, change the ASPM Control entry in Table 7-15 as follows:

Table 7-15: Link Control Register

Bit Location	Register Description	Attributes
1: 0	**Active State Power Management (ASPM) Control** – This field controls the level of ASPM supported on the given PCI Express Link. Defined encodings are: 00b Disabled 01b L0s Entry Enabled 10b L1 Entry Enabled 11b L0s and L1 Entry Enabled Note: "L0s Entry Enabled" indicates the Transmitter entering L0s is supported. The Receiver must be capable of entering L0s even when the field is disabled (00b). Default value of this field is ~~(~~00b~~)~~ ~~or (01b) depending on~~unless otherwise required by a particular form factor.	RW
…	…	…
7	**Extended Sync** – This bit when set forces the transmission of ~~4096 FTS~~additional ordered sets ~~in~~ when exiting the L0s state ~~followed by a single SKP ordered set~~ (see Section 4.2.4.3) ~~prior to entering the L0 state, and the transmission of 1024 TS1 ordered sets in the L1 state prior to entering~~ and when in the Recovery state (see Section 4.2.6.4.1). This mode provides external devices (e.g., logic analyzers) monitoring the Link time to achieve bit and Symbol lock before the Link enters the L0 ~~or Recovery~~ state~~s~~ and resumes communication. Default value for this bit is 0b.	RW

C35. Zero Length Read Implementation Note

PWG Approved 20 May 2004 -- Release Date: 12 Jul 2004

Edit as shown:

2.2.5. First/Last DW Byte Enables Rules

…

IMPLEMENTATION NOTE

Zero Length Read

A Memory Read Request of 1 DW with no bytes enabled, or "zero length Read," may be used by devices as a type of flush Request. For a Requester, the flush semantic allows a device to ensure that previously issued Posted Writes have been completed at their PCI Express destination. To be effective in all cases, the address for the zero-length Read must target the same device as the Posted Writes that are being flushed. One recommended approach is using the same address as one of the Posted Writes being flushed.

C36. Miscellaneous Errata

PWG Approved 24 Jun 2004 -- Release Date: 12 Jul 2004

In Section 7.8.7. Link Control Register:

4	**Link Disable** – This bit disables the Link when set to 1b; this field is not applicable and reserved for Endpoint devices, PCI Express PCI/PCI-X bridges and Upstream Ports of Switches. Writes to this bit are immediately reflected in the value read from the bit, regardless of actual Link state. Default value of this field is 0b.	RW
5	**Retrain Link –** A write of 1b to this bit initiates Link retraining by directing the Physical Layer LTSSM to the Recovery state. Reads of this bit always return 0b. This field is not applicable and is reserved for Endpoint devices, PCI Express to PCI/PCI-X bridges and Upstream Ports of Switches. This bit always returns 0b when read.	RW

In Section 2.6.1.1:

- A Receiver must never cumulatively issue more than ~~2048~~ 2047 outstanding unused credits to the Transmitter for data payload or ~~128~~ 127 for header.
  - Components may optionally check for violations of this rule. If a component implementing this check determines a violation of this rule, the violation is a Flow Control Protocol Error (FCPE).
    - If checked, this is a reported error associated with the Receiving Port (see Section 6.2)

In Section 5.3.1.2.:

D1 support is optional. While in the D1 state, a function must not initiate any TLPs on the Link with the exception of a PME Message as defined in Section 5.3.3. Configuration and Message requests are the only

TLPs accepted ~~(as target)~~ by a function ~~that is currently~~ in the D1 state. All other received Requests must be handled as Unsupported Requests, and all received Completions must be handled as Unexpected Completions. If an error caused by a received TLP (e.g. an Unsupported Request) is detected while in D1, and reporting is enabled, the link must be returned to L0 if it is not already in L0 and an error message must be sent. If an error caused by an event other than a received TLP (e.g. a Completion Timeout) while in D1, an error message must be sent when the device is programmed back to the D0 state.

...

In Section 5.3.1.3.:

D2 support is optional. While in the D2 state, a function must not initiate any TLPs on the Link with the exception of a PME Message as defined in Section 5.3.3. Configuration and Message requests are the only TLPs accepted ~~(as target)~~ by a function ~~that is currently~~ in the D2 state. All other received ~~TLPs~~ Requests must be handled as ~~unsupported~~ Unsupported ~~packets~~ Requests~~.~~, and all received Completions must be handled as Unexpected Completions. If an error caused by a received TLP (e.g. an Unsupported Request) is detected while in D2, and reporting is enabled, the link must be returned to L0 if it is not already in L0 and an error message must be sent. If an error caused by an event other than a received TLP (e.g. a Completion Timeout) while in D2, an error message must be sent when the device is programmed back to the D0 state.

...

In Section 5.3.1.4.1.:

Configuration and Message requests are the only TLPs accepted by a function in the $D3_{hot}$ state. ~~When a function is in $D3_{hot}$, it must respond to configuration accesses targeting it. They~~ Functions in the $D3_{hot}$ state must ~~also~~ participate in the PME_Turn_Off/PME_TO_Ack protocol. Refer to Section 5.3.3 details. All other received Requests may optionally be handled as Unsupported Requests, and all received Completions may optionally be handled as Unexpected Completions. If an error caused by a received TLP (e.g. an Unsupported Request) is detected while in $D3_{hot}$, and reporting is enabled, the link must be returned to L0 if it is not already in L0 and an error message must be sent. If an error caused by an event other than a received TLP (e.g. a Completion Timeout) while in $D3_{hot}$, an error message may optionally be sent when the device is programmed back to the D0 state. Once in $D3_{hot}$ the function can later be transitioned into $D3_{cold}$ (by removing power from its host component).

...

In Table 7-8 (Building on errata C19):

22	**B2/B3 Support** ~~– Does not apply to PCI Express. Must be hardwired to 0.~~	Unchanged
23	**Bus Power/Clock Control Enable** ~~– Does not apply to PCI Express. Must be hardwired to 0.~~	Unchanged

In Section 2.2:

...

All TLP fields marked Reserved (sometimes abbreviated as R) must be filled with all 0's when a TLP is formed. Values in such fields must be ignored by Receivers and must be forwarded unmodified by Switches. Note that for certain fields there are both specified and reserved values – the handling of reserved values in these cases is specified separately for each case.

...

In Figure 7-38 Change:

"VAT_Offset * ~~04h~~10h" for VC Arbitration Table

"PAT_Offset * ~~04h~~10h" for Port Arbitration Table

In 7.5.3.3 - Change as indicated (building on earlier errata):

14	**Received System Error** – See Section 7.5.1.7. This bit is set when the Secondary Side for a Type 1 configuration space header device receives an ERR_FATAL or ERR_NONFATAL Message~~, and the SERR# Enable bit in the Bridge Control register is 1b~~. Default value of this field is 0.	RW1C

C37. Withdrawn [was: Completion ID-Based Routing]

C38. Deleted [Duplicate]

C39. Deleted [Duplicate]

C40. Message Header Bytes 8 -15

PWG Approved 24 Jun 2004 -- Release Date: 12 Jul 2004

In Section 2.2.8. Message Request Rules, edit as shown:

...

- The Message Code field must be fully decoded (Message aliasing is not permitted).
- Except as noted, the Attr[1:0] field is reserved.
- Except as noted, bytes 8 through 15 are reserved.
- Message Requests are posted and do not require Completion.
- Message Requests follow the same ordering rules as Memory Write Requests.

Figure 2-17: Message Request Header ***[Change label for bytes 8-15 (grey) to: "Except as noted, bytes 8 through 15 are reserved"]***

In addition to address and ID routing, Messages support several other routing mechanisms. These mechanisms are referred to as "implicit" because no address or ID specifies the destination device, but rather the destination is implied by the routing type. The following rules cover Message routing mechanisms:

- Message routing is determined using the r[2:0] sub-field of the Type field
    - Message Routing r[2:0] values are defined in Table 2-11
    - Permitted values are defined in the following sections for each Message

Table 2-11: Message Routing

r[2:0]	Description	Bytes 8 Through 15[7]
000	Routed to Root Complex	Reserved
001	Routed by Address[8]	Address
010	Routed by ID[9]	See Section 2.2.4
011	Broadcast from Root Complex	Reserved
100	Local - Terminate at Receiver	Reserved
101	Gathered and routed to Root Complex[10]	Reserved
110-111	Reserved - Terminate at Receiver	Reserved

[7] Except as noted, eg Vendor Defined Messages

[8] Note that no Messages defined in this document use Address routing.

[9] ~~See Section 2.2.4.~~

[10] This routing type is used only for PME_TO_Ack, and is described in Section 0.

C41. Crosslink

EWG Approved 10 Jun 2004 -- Release Date: 12 Jul 2004

In Section 4.2.6.3.1.2. – line 5, page 178

- Otherwise, after a Tcrosslink timeout, 16-32 TS2 ordered sets with PAD link numbers and PAD lane numbers are sent. The ~~the~~ Upstream Lanes become Downstream Lanes and the next state is Configuration.Linkwidth.Start as Downstream Lanes.
    - Note: This optional behavior is required for crosslink behavior where two Ports may start off with Upstream Ports, and one will eventually take the lead as a Downstream Port.

Insert note in Section 4.2.6.3.1.1. – line 22, page 176

- Optionally, if crosslinks are supported and all Downstream Lanes initially receive two consecutive TS1 ordered sets with a Link number different than PAD (K23.7) and a Lane Number set to PAD, the Downstream Lanes are now designated as Upstream Lanes and a new random cross Link timeout is chosen (see Tcrosslink in Table 4-5). The next state is Configuration.Linkwidth.Start as Upstream Lanes.
    - Note: This supports the optional crosslink where both sides may try to act as a DownstreamPort. This is resolved by making both Ports become Upstream and assigning a random timeout until one side of the Link becomes a Downstream Port and the other side remains an Upstream Port. This timeout must be random even when hooking up two of the same devices so as to eventually break any possible deadlock.
    - Note: If crosslinks are supported, receiving a sequence of TS1 ordered sets with a Link number of PAD followed by a Link number of non-PAD that matches the transmitted link number is only valid when not interrupted by the reception of a TS2 ordered set.

C42. Error Reporting and Logging

PWG Approved 1 Jul 2004 -- Release Date: 12 Jul 2004

The following errata include changes to the Root Complex Integrated Devices Event Collector ECN to the PCI Express Base Specification, Rev 1.0a.

In Section 6.2.4.1, edit/add text as shown ("base" text includes the RC IE/EC ECN):

In addition to the above logging, a Root ~~Complex~~ Port or a Root Complex Event Collector that supports the Advanced Error Reporting capability is required to implement the Error Source Identification register, which records the Requester ID of the first ERR_NONFATAL/ERR_FATAL (uncorrectable errors) and ERR_COR (correctable errors) Messages received by the Root Port or Root Complex Event Collector~~Complex~~. System software written to support Advanced Error Reporting can use the Root ~~Port~~ Error Status register to determine which fields hold valid information.

If a Root Complex Event Collector is implemented, errors from a Root Complex Integrated Endpoint may optionally be reported in a Root Complex Event Collector residing on the same logical bus as the Root Complex Integrated Endpoint. The Root Complex Event Collector must explicitly declare supported Root Complex Integrated Endpoints as part of its capabilities. Each Root Complex Integrated Endpoint must be associated with exactly one Root Complex Event Collector.

For both Root Ports and Root Complex Event Collectors, in order for a received error Message or an internally generated error Message to be recorded in the Root Error Status register and the Error Source Identification register, the error Message must be "transmitted". See Section 6.2.7.1 for information on how received Messages are forwarded and transmitted. Internally generated error Messages are enabled for transmission with the SERR# Enable bit in the Command Register (ERR_NONFATAL and ERR_FATAL) or the Reporting Enable bits in the Device Control Register (ERR_COR, ERR_NONFATAL, and ERR_FATAL).

In Section 6.2.7.1, add bullet as shown (Errata C2 applied to existing text for readability):

….

- ERR_COR, ERR_NONFATAL and ERR_FATAL are forwarded from the secondary interface to primary interface if the SERR# Enable bit in the Bridge Control register is set. Transmission of forwarded ERR_NONFATAL and ERR_FATAL messages by the primary interface is controlled by the SERR# Enable bit in the Command register.
- For a Root Port, error Messages forwarded from the secondary interface to the primary interface must be enabled for "transmission" by the primary interface in order to cause a System Error via the Root Control register or (when the Advanced Error Reporting Capability is present) reporting via the Root Error Command Register and logging in the Root Error Status Register and Error Source Identification Register
- For a Root Complex Event Collector (technically not a Bridge), error Messages "received" from associated Root Complex Integrated Endpoints must be enabled for "transmission" in order to cause a System Error via the Root Control register or (when the Advanced Error Reporting Capability is present) reporting via the Root Error Command Register and logging in the Root Error Status Register and Error Source Identification Register. Since a Root Complex Event Collector has no Bridge Control register, ERR_COR is always enabled for "transmission", while ERR_NONFATAL and ERR_FATAL are enabled for "transmission" by the SERR# Enable bit in the Command register.

In Section 7.10.9, add text as shown ("base" text includes the RC IE/EC ECN):

The Root Error Command register allows further control of Root Complex response to Correctable, Non-Fatal, and Fatal error Messages than the basic Root Complex capability to generate system errors in response to error Messages. Bit fields (see Figure 7-35) enable or disable generation of interrupts (claimed by the Root Port or Root Complex Event Collector) in addition to system error Messages according to the definitions in Table 7-31.

For both Root Ports and Root Complex Event Collectors, in order for a received error Message or an internally generated error Message to generate an interrupt enabled by this register, the error Message must be enabled for "transmission" by the Root Port or Root Complex Event Collector. See Sections 6.2.4.1 and 6.2.7.1.

In Section 7.10.10, add text as shown ("base" text includes the RC IE/EC ECN):

The Root Error Status register reports status of error Messages (ERR_COR~~)~~, ERR_NONFATAL, and ERR_FATAL) received by the Root ~~Complex~~Port, and of errors detected by the Root Port itself (which are treated conceptually as if the Root Port had sent an error Message to itself). In order to update this register, error Messages received by the Root Port and/or internally generated error Messages must be enabled for

"transmission" by the primary interface of the Root Port. ERR_NONFATAL and ERR_FATAL Messages are grouped together as uncorrectable. ...

…

Root Complex Event Collectors provide support for the above described functionality for Root Complex Integrated Endpoints (and for the Root Complex Event Collector itself). In order to update this register, error Messages received by the Root Complex Event Collector fromits associated Root Complex Integrated Endpoints and/or internally generated error Messages must be enabled for "transmission" by the Root Complex Event Collector.

C43. Max_Link_Width

PWG Approved 24 Jun 2004 -- Release Date: 12 Jul 2004

Edit as shown in 7.8.6. Link Capabilities Register (Offset 0Ch):

...

Table 7-14: Link Capabilities Register

Bit Location	Register Description	Attributes
...		
9:4	**Maximum Link Width** – This field indicates ~~the maximum width of the given PCI Express Link.~~ the maximum link width (xN – corresponding to N lanes) implemented by the component. This value is permitted to exceed the number of lanes routed to the slot (Downstream Port), adapter connector (Upstream Port), or in the case of component to component connections, the actual wired connection width. Defined encodings are: 000000b Reserved 000001b x1 000010b x2 000100b x4 001000b x8 001100b x12 010000b x16 100000b x32	RO
...		

C44. Electrical Idle Exit

EWG Approved 10 Jun 2004 -- Release Date: 12 Jul 2004

In 4.3.1.9. edit as shown:

Electrical Idle Exit occurs ~~when a signal larger~~ at a threshold voltage no less than the minimum $V_{RX\text{-}IDLE\text{-}DET\text{-}DIFFp\text{-}p}$ ~~is detected at a Receiver~~ limit and no greater than the maximum $V_{RX\text{-}IDLE\text{-}DET\text{-}DIFFp\text{-}p}$.

C45. DC Impedance

EWG Approved 8 Jul 2004 -- Release Date: 13 Jul 2004

Remove parameter ZTX-DC from the specification:

- In section 4.2.4.5.1 (page 179), replace “The Transmitter terminations are required to only meet $Z_{TX\text{-}DC}$ (see Table 4-11)” with “The Transmitter is required only to meet $I_{TX\text{-}SHORT}$ (see Table 4-9).”
- In section 4.3.1.4 (Page 223), replace “The Transmitter is required to only meet $Z_{TX\text{-}DC}$ (see Table 4-9) anytime functional differential signals are not being transmitted” with “The Transmitter is required only to meet $I_{TX\text{-}SHORT}$ (see Table 4-9) anytime functional differential signals are not being transmitted”
- In section 4.3.3 (Table 4-9 on page 234), completely remove the row with $Z_{TX\text{-}DC}$.

Modify parameter ZTX-DIFF-DC :

- In section 4.3.3 (Table 4-9 on page 234), in the comments column for $Z_{TX\text{-}DIFF\text{-}DC}$, replace “TX DC Differential Mode Low impedance” with “TX DC Differential Mode Low impedance. See Note 6.”
- At the end of Table 4-9 on page 235, add the following note (numbered 6):

> “$Z_{TX\text{-}DIFF\text{-}DC}$ is the small signal resistance of the transmitter measured at a DC operating point that is equivalent to that established by connecting a 100ohm resistor from D+ and D- while the TX is driving a static logic one or logic zero. Equivalently, this parameter can be derived by measuring the RMS voltage of the TX while transmitting a test pattern into two different differential terminations that are near 100 ohms.
>
> Small signal resistance is measured by forcing a small change in differential voltage and dividing this by the corresponding change in current.”

C46. MSI/MSI-X

SWWG Approved 30 Jun 2004 -- Release Date: 12 Jul 2004

Note: This errata builds on errata C21 and the MSI-X ECN.

6.1.4. Message Signaled Interrupt (MSI) Support, change text as shown:

6.1.4. Message Signaled Interrupt (MSI/MSI-X) Support

~~The Message Signaled Interrupt (MSI) capability is defined in the PCI 2.3 Specification.~~

MSI/MSI-X interrupt support, which is optional for PCI 2.3 devices, is required for PCI Express devices. All PCI Express devices that are capable of generating interrupts must support MSI or MSI-X or both. ~~MSI-capable devices~~ The MSI and MSI-X mechanisms deliver interrupts by performing memory write transactions. MSI and MSI-X ~~is~~ are ~~an~~ edge-triggered interrupt mechanisms; neither the PCI 2.3 Specification nor this specification support level-triggered MSI/MSI-X interrupts. Certain PCI devices and their drivers rely on INTx-type level-triggered interrupt behavior (addressed by the PCI Express legacy INTx emulation mechanism). To take advantage of the MSI or MSI-X capability and edge-triggered interrupt semantics, these devices and their drivers may have to be redesigned.

IMPLEMENTATION NOTE

Per-Vector Masking with MSI/MSI-X

Devices and drivers that use MSI or MSI-X have the challenge of coordinating exactly when new interrupt messages are generated. If hardware fails to send an interrupt message that software expects, an interrupt event might be "lost". If hardware sends an interrupt message that software is not expecting, a "spurious" interrupt might result.

Per-Vector Masking (PVM) is an architected feature for MSI and MSI-X that can be used to assist in this coordination. For example, when a software interrupt service routine begins, it can mask the vector to help avoid "spurious" interrupts. After the interrupt service routine services all the interrupt conditions that it is aware of, it can unmask the vector. If any interrupt conditions remain, hardware is required to generate a new interrupt message, guaranteeing that no interrupt events are lost.

PVM is a standard feature with MSI-X and an optional feature for MSI. For devices that implement MSI, implementing PVM as well is highly recommended.

6.1.5 Native PME Software Model, change text as shown:

6.1.5 Native PME Software Model

PCI Express-aware software can enable a mode where the Root Complex signals PME via an interrupt. When configured for native PME support, a Root Port receives the PME Message and sets the PME Status bit in its Root Status register. If software has set the PME Interrupt Enable bit in the Root Control register to 1b, the Root Port then generates an interrupt.

If the Root Port is enabled for level-triggered interrupt signaling using the INTx messages, the virtual INTx wire must be asserted whenever and as long as the following conditions are all satisfied:

- The Interrupt Disable bit in the Command register is set to 0b
- The PME Interrupt Enable bit in the Root Control register is set to 1b
- The PME Status bit in the Root Status register is set

Note that all other interrupt sources within the same Function will assert the same virtual INTx wire when requesting service.

If the Root Port is enabled for edge-triggered interrupt signaling using MSI or MSI-X, an interrupt message must be sent every time the logical AND of the following conditions transitions from FALSE to TRUE:

- The associated vector is unmasked (not applicable if MSI does not support PVM)
- The PME Interrupt Enable bit in the Root Control register is set to 1b

- The PME Status bit in the Root Status register is set

Note that PME and Hot-Plug Event interrupts (when both are implemented) always share the same MSI or MSI-X vector, as indicated by the Interrupt Message Number field in the PCI Express Capabilities register.

6.2.4.1. Root Complex Considerations (Advanced Error Reporting), change text as shown:

6.2.4.1. Root Complex Considerations (Advanced Error Reporting)

6.2.4.1.1 Error Source Identification

In addition to the above logging, a Root Complex that supports the Advanced Error Reporting capability is required to implement the Error Source Identification register, which records the Requester ID of the first ERR_NONFATAL/ERR_FATAL (uncorrectable errors) and ERR_COR (correctable errors) Messages received by the Root Complex. System software written to support Advanced Error Reporting can use the Root Error Status register to determine which fields hold valid information.

6.2.4.1.2 Interrupt Generation

The Root Error Command register allows further control of Root Complex response to Correctable, Non-Fatal, and Fatal error Messages than the basic Root Complex capability to generate system errors in response to error Messages. Bit fields enable or disable generation of interrupts for the three types of error Messages. System error generation in response to error Messages may be disabled via the PCI Express Capability structure.

If a Root Port or Root Complex Event Collector is enabled for level-triggered interrupt signaling using the INTx messages, the virtual INTx wire must be asserted whenever and as long as the following conditions are all satisfied:

- The Interrupt Disable bit in the Command register is set to 0b
- At least one error Message Reporting Enable bit in the Root Error Command register and its associated error Messages Received bit in the Root Error Status register are both set to 1b

Note that all other interrupt sources within the same Function will assert the same virtual INTx wire when requesting service.

If a Root Port or Root Complex Event Collector is enabled for edge-triggered interrupt signaling using MSI or MSI-X, an interrupt message must be sent every time the logical AND of the following conditions transitions from FALSE to TRUE:

- The associated vector is unmasked (not applicable if MSI does not support PVM)
- At least one error Message Reporting Enable bit in the Root Error Command register and its associated error Messages Received bit in the Root Error Status register are both set to 1b

Note that Advanced Error Reporting MSI/MSI-X interrupts always use the vector indicated by the Advanced Error Interrupt Message Number field in the Root Error Status register.

Note: The following section modifies text introduced by C21 (referred to there as section 6.7.7.3)

6.7.5.3. Software Notification of Hot-Plug Events, change text as shown:

6.7.5.3. Software Notification of Hot-Plug Events

A hot-plug capable Downstream Port must support generation of an interrupt on a hot-plug event. As described in Sections 6.7.5.1 and 6.7.5.2, each hot-plug event has both an enable bit for interrupt generation and a status bit that indicates when an event has occurred but has not yet been processed by software. There is also a Hot-Plug Interrupt Enable bit in the Slot Control register that serves as a master enable/disable bit for all hot-plug events.

~~As described in Sections 6.7.5.1 and 6.7.5.2, each type of hot-plug event has both an enable field which enables software notification of that type of event and a status field that indicates that an event has occurred but has not yet been processed by software.~~

~~If the enable field for an event indicates that software notification of the event is disabled, the hotplug capable Downstream Port must not generate an interrupt on that event. Hot-plug interrupt generation is also globally enabled through the Hot-Plug Interrupt Enable field in the Slot Control register. If this field indicates that hot-plug interrupts are disabled, the hot-plug capable Downstream Port must not generate an interrupt on any hot-plug event.~~

~~The Downstream Port must generate an interrupt on a hot-plug event following the rules for interrupt generation described in the rest of this document provided these disable conditions are not present and the corresponding status field for the event is not set. If the status field is set, software has not yet handled a previous event; thus, a new interrupt must not be generated.~~

~~If the port is enabled for level-triggered interrupt generation using the INTx messages, the interrupt must remain asserted as long as global interrupt generation is enabled and at least one status field for an enabled hot-plug event remains set.~~If the port is enabled for level-triggered interrupt signaling using the INTx messages, the virtual INTx wire must be asserted whenever and as long as the following conditions are all satisfied:

- The Interrupt Disable bit in the Command register is set to 0b
- The Hot Plug Interrupt Enable bit in the Slot Control register is set to 1b
- At least one hot-plug event status bit in the Slot Status register and its associated Enable bit in the Slot Control register are both set to 1b

Note that all other interrupt sources within the same Function will assert the same virtual INTx wire when requesting service.

If the port is enabled for edge-triggered interrupt ~~generation~~signaling using MSI or MSI-X, ~~an MSI must be sent whenever global interrupt generation is enabled and the status field for an enabled hot-plug event transitions from not set to set.~~an interrupt message must be sent every time the logical AND of the following conditions transitions from FALSE to TRUE:

- The associated vector is unmasked (not applicable if MSI does not support PVM)
- The Hot Plug Interrupt Enable bit in the Slot Control register is set to 1b
- At least one hot-plug event status bit in the Slot Status register and its associated Enable bit in the Slot Control register are both set to 1b

Note that PME and Hot-Plug Event interrupts (when both are implemented) always share the same MSI or MSI-X vector, as indicated by the Interrupt Message Number field in the PCI Express Capabilities register.

7.8.2. PCI Express Capabilities Register, change text as shown:

Table 7-10: PCI Express Capabilities Register

Bit Location	Register Description	Attributes
…	**…**	…

Bit Location	Register Description	Attributes
13:9	**Interrupt Message Number** – ~~If this function is allocated more than one MSI interrupt number, this register is required to contain the offset between the base Message Data and the MSI Message that is~~ This register must indicate which MSI/MSI-X vector is used for the interrupt message generated ~~when any of~~in association with the status bits in either the Slot Status register (see section 6.7.5.3) or the Root ~~Port~~ Status register (see Section 6.1.6) of this capability structure ~~are set~~. …	RO

7.8.12. Root Control Register, change text as shown:

Table 7-39: Root Control Register

Bit Location	Register Description	Attributes
…	**…**	RW
3	**PME Interrupt Enable** – This bit when set enables interrupt generation upon receipt of a PME Message as reflected in the PME Status register bit (see Table 7-42). A PME interrupt is also generated if the PME Status register bit is set when this bit is set from a cleared state. See Section 6.1.5. Default value of this field is 0.	RW

7.10.10. Root Error Status Register, p. 376, change text as shown:

Table 7-32: Root Error Status Register

Bit Location	Register Description	Attributes
…	**…**	…
31:27	**Advanced Error Interrupt Message Number** – ~~If this function is allocated more than one MSI interrupt number, this register is required to contain the offset between the base Message Data and the MSI Message that is~~ This register must indicate which MSI/MSI-X vector is used for the interrupt message generated ~~when~~ in association with any of the status bits of this capability ~~are set~~. …	RO

C47. Version Number

SWWG Approved 6 July 2004 -- Release Date: 12 Jul 2004

In 7.8.2 PCI Express Capabilities Register (Offset 02h):

The PCI Express Capabilities register identifies PCI Express device type and associated capabilities. Figure 7-12 details allocation of register fields in the PCI Express Capabilities register; Table 7-10 provides the respective bit definitions.

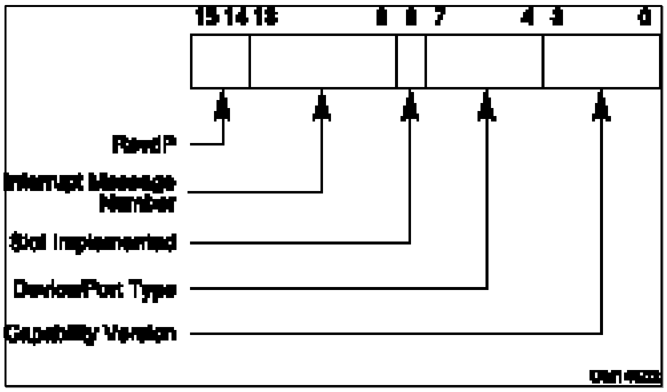


Figure 7-12: PCI Express Capabilities Register

Table 7-10: PCI Express Capabilities Register

Bit Location	Register Description	Attributes
3:0	**Capability Version** – Indicates PCI-SIG defined PCI Express capability structure version number. <u>A version of the specification that changes the PCI Express capability structure in a way that is not otherwise identifiable (e.g. through a new capability field) is permitted to increment this field. All such changes to the PCI Express capability structure must be software-compatible. Software must check for Capability Version numbers that are greater than or equal to the highest number defined when the software is written, as devices reporting any such Capability Version numbers will contain a PCI Express capability structure that is compatible with that piece of software.</u> Must be 1h for <u>devices compliant to </u>this specification.	RO

…

In 7.9.3 PCI Express Enhanced Capability Header:

All PCI Express extended capabilities must begin with a PCI Express Enhanced Capability header. Figure 7-25 details the allocation of register fields of a PCI Express Extended Capability header; Table 7-22 provides the respective bit definitions.

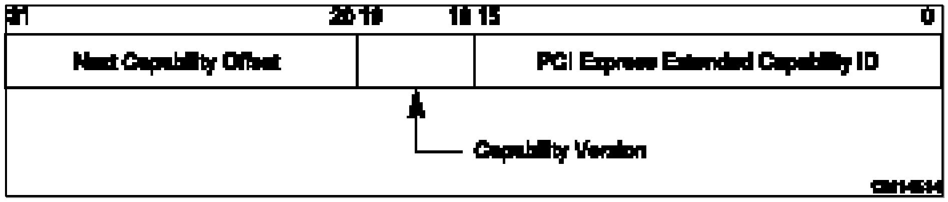


Figure 7-25: PCI Express Enhanced Capability Header

Table 7-22: PCI Express Enhanced Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the extended capability.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the capability structure present. A version of the specification that changes the extended capability in a way that is not otherwise identifiable (e.g. through a new capability field) is permitted to increment this field. All such changes to the capability structure must be software-compatible. Software must check for Capability Version numbers that are greater than or equal to the highest number defined when the software is written, as devices reporting any such Capability Version numbers will contain a capability structure that is compatible with that piece of software.	RO
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express capability structure or 000h if no other items exist in the linked list of capabilities. For Extended Capabilities implemented in device configuration space, this offset is relative to the beginning of PCI compatible configuration space and thus must always be either 000h (for terminating list of capabilities) or greater than 0FFh. The bottom two bits of this offset are reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits.	RO

C48. (E1) Clarify Electrical Idle Exit in text

Posting Date: 9/30/04

Replace the last sentence section 4.3.1.9 (page 225)

DELETE:

Electrical Idle Exit occurs when a signal at a threshold voltage no less than the minimum $V_{RX\text{-}IDLE\text{-}DET\text{-}DIFFp\text{-}p}$ limit and no greater than the maximum $V_{RX\text{-}IDLE\text{-}DET\text{-}DIFFp\text{-}p}$.

REPLACE WITH:

Electrical Idle Exit shall not occur if a signal smaller than $V_{RX\text{-}IDLE\text{-}DET\text{-}DIFFp\text{-}p}$ minimum is detected at a Receiver. Electrical Idle Exit shall occur if a signal larger than $V_{RX\text{-}IDLE\text{-}DET\text{-}DIFFp\text{-}p}$ maximum is detected at a Receiver.

C49. (E2) Clarify Clock recovery function

Posting Date: 9/30/04

Replace 2 sentences in the Comments column for table 4-11 (RX input spec), $T_{RX\text{-}EYE\text{-}MEDIAN\text{-}TO\text{-}MAX\text{-}JITTER}$ symbol row (page 237):

DELETE:
A recovered TX UI is calculated over 3500 consecutive unit intervals of sample data. Jitter is measured using all edges of the 250 consecutive UI in the center of the 3500 UI used for calculating the TX UI.

REPLACE WITH:
To be measured after the clock recovery function in Section 4.3.3.2.

C50. (E3) Fix wrong symbol names

Posting Date: 9/30/04

Replace symbol names in section 4.3.1.4 (page 222). The symbols name should be consistent with tables4-9, 4-11.

REPLACE: $RL_{TX\text{-}DM}$ WITH: $RL_{TX\text{-}DIFF}$
DELETE A SYMBOL NO LONGER EXIST: $Z_{TX\text{-}DC}$
REPLACE: $RL_{RX\text{-}DM}$ WITH: $RL_{RX\text{-}DIFF}$

C51. (E4) Update VNA differential test input signal level

Posting Date: 10/1/04

To recap the issue which was brought before the group yesterday, the footnotes specific to the amplitude of the stimulus required for the Transmitter and Receiver differential/common mode return loss is a bit high in magnitude. To satisfy the signal levels commonly found in TDNA classes of equipment, I am requesting changing the wording around 300mV RMS to say 200mV Peak.

Section 4.3.3 page 233 (16Jul04RD) states..
------Current Text------
4. The Transmitter input impedance shall result in a differential return loss greater than or equal to 10 dB with differential test input signals of 300 mV (RMS value) swing around ground applied to D+ and D- lines and a common mode return loss greater than or equal to 6 dB over a

frequency range of 50 MHz to 1.25 GHz. This input impedance requirement applies to all valid input levels. The reference impedance for return loss measurements is 50 ohms to ground for both the D+ and D- line (i.e., as measured by a Vector Network Analyzer with 50 ohm probes - see Figure 4-49). Note that the series capacitors C_{TX} is optional for the return loss measurement.

Proposed revision.

4. The Transmitter input impedance shall result in a differential return loss greater than or equal to 10 dB with a differential test input signal no less than 200 mV (peak value, 400mV differential peak to peak) swing around ground applied to D+ and D- lines and a common mode return loss greater than or equal to 6 dB over a frequency range of 50 MHz to 1.25 GHz. This input impedance requirement applies to all valid input levels. The reference impedance for return loss measurements is 50 ohms to ground for both the D+ and D- line (i.e., as measured by a Vector Network Analyzer with 50 ohm probes - see Figure 4-49). Note that the series capacitors C_{TX} is optional for the return loss measurement.

Section 4.3.4, table 4-11, page 236 (16Jul04RD) states,

$RL_{RX\text{-}DIFF}$

------Current Text------

Measured over 50 MHz to 1.25 GHz with differential test input signals of 300 mV (RMS value) swing around ground applied to the D+ and D- lines. See Note 9.

Proposed revision.

Measured over 50 MHz to 1.25 GHz. See Note 11. << note the footnote is now #11

$RL_{RX\text{-}CM}$

------Current Text------

Measured over 50 MHz to 1.25 GHz with differential test input signals of 300 mV (RMS value) swing around ground applied to the D+ and D- lines. See Note 9.

Proposed revision.

Measured over 50 MHz to 1.25 GHz. See Note 11. << note the footnote is now #11

------Current Text------

11. The Receiver input impedance shall result in a differential return loss greater than or equal to 10 dB with differential test input signals of 300 mV (RMS value) swing around ground applied to D+ and D- lines and a common mode return loss greater than or equal to 6 dB (no bias required) over a frequency range of 50 MHz to 1.25 GHz. This input impedance requirement applies to all valid input levels. The reference impedance for return loss measurements for is 50 ohms to ground for both the D+ and D- line (i.e., as measured by a Vector Network Analyzer with 50 ohm probes - see Figure 4-49). Note: that the series capacitors C_{TX} is optional for the return loss measurement.

Proposed revision.

11. The Receiver input impedance shall result in a differential return loss greater than or equal to 10 dB with a differential test input signal of no less than 200 mV (peak value, 400mV differential peak to peak) swing around ground applied to D+ and D- lines and a common mode return loss greater than or equal to 6 dB (no bias required) over a frequency range of 50 MHz to 1.25 GHz. This input impedance requirement applies to all valid input levels. The reference impedance for

return loss measurements for is 50 ohms to ground for both the D+ and D- line (i.e., as measured by a Vector Network Analyzer with 50 ohm probes - see Figure 4-49). Note: that the series capacitors C_{TX} is optional for the return loss measurement.

------Current Text------
13. The RX DC Common Mode Impedance that exists when no power is present or Fundamental Reset is asserted. This helps ensure that the Receiver Detect circuit will not falsely assume a Receiver is powered on when it is not. This term must be measured at 300 mV above the RX ground.

Proposed revision.
13. The RX DC Common Mode Impedance that exists when no power is present or Fundamental Reset is asserted. This helps ensure that the Receiver Detect circuit will not falsely assume a Receiver is powered on when it is not. This term must be measured at 200 mV above the RX ground.

C52. (E5) Clarify RX DC common mode allowance

Posting Date: 10/8/04

```
There was a feedback regarding an inconsistency between PCI Express
Endpoint Checklist vs. Base Spec about RX DC Common Mode Voltage:

Checklist states: 0 volts +/- 10mV
Base Spec states: always 0 V

Here's a proposal to keep the Checklist consistent with Spec.

Section 4.3.1.5:

---current text----
The Receiver DC common mode voltage is always 0 V during all states.

---proposed text---
The Receiver DC common mode voltage is nominally 0 V during all
states..
```

C53. (E6) Clarify requirement for detect electrical idle in L0

Posting Date: 10/8/04

Section 4.2.6.5, third bullet:

Change: "Next state is Recovery if directed to this state or if Electrical Idle is detected on all Lanes without receiving an Electrical Idle ordered set on any lane"

To: "Next state is Recovery if directed to this state; If Electrical Idle is detected on all lanes without receiving an Electrical Idle ordered set on any lane, the port may transition to the Recovery state or may remain in L0. In the event that the port is in L0 and the Electrical Idle condition occurs without receiving an Electrical Idle Ordered Set, errors may occur and the port may be directed to transition to Recovery.

C54. (E7) Clarify clean reference clock

Posting Date: 10/13/04

Section 4.3.3, Table 4-9, row 4 column 7 (comments).

Replace

"This parameter is measured with a clean reference clock"

with

"This parameter is measured with the equivalent of a zero jitter reference clock"

Same table note 3, replace

This is to be measured with a clean reference clock, such as a high quality function generator.

With

This parameter is measured with the equivalent of a zero jitter reference clock.

In section 4.3.4 table 4-11 note 9, replace

This is to be measured with a clean reference clock applied to the transmitter, such as a high quality function generator

With

This parameter is measured with the equivalent of a zero jitter reference clock.

C55. (E8) Clarify transfer function

Posting Date: 10/13/04

On the top page of 233, change equation 4.3 from:

$$H_d(w) = \left| \left(\frac{s}{s + w_3} \right)^3 \right|$$

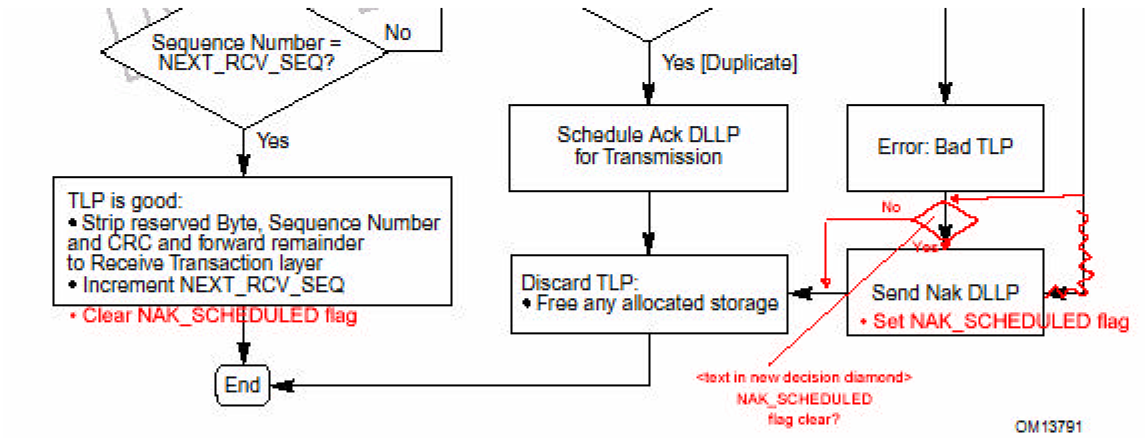
to

$$H_d(w) = |\, 2*\left(\frac{s}{s+w_3}\right)^3 \,|$$

C56. (Cx1) Miscellaneous Errata on the 1.1RD

PWG Approved 14 Oct 2004 -- Release Date: 28 Oct 2004

In Figure 3-34, edit as shown (note: unchanged parts of figure not shown):



Section 5.3.1.2 D1 State, Page 248 line 16:

5.3.1.2. D1 State

D1 support is optional. While in the D1 state, a function must not initiate any Request TLPs on the Link with the exception of a PME Message as defined in Section 5.3.3. Configuration and Message requests are the only TLPs accepted by a function in the D1 state. All other received Requests must be handled as Unsupported Requests, and all received Completions ~~must~~ may optionally be handled as Unexpected Completions. If an error caused by a received TLP (e.g., an Unsupported Request) is detected while in D1, and reporting is enabled, the link must be returned to L0 if it is not already in L0 and an error message must be sent. If an error caused by an event other than a received TLP (e.g,. a Completion Timeout) is detected while in D1, an error message must be sent when the device is programmed back to the D0 state..

Section 5.3.1.3 D2 State, Page 249 line 1:

5.3.1.3. D2 State

D2 support is optional. While in the D2 state, a function must not initiate any Request TLPs on the Link with the exception of a PME Message as defined in Section 5.3.3. Configuration and Message requests are the only TLPs accepted by a function in the D2 state. All other received Requests must be handled as Unsupported Requests, and all received Completions ~~must~~ may optionally be handled as Unexpected Completions. If an error caused by a received TLP (e.g., an Unsupported Request) is detected while in D2, and reporting is enabled, the link must be returned to L0 if it is not already in L0 and an error message must be sent. If an error caused by an event other than a received TLP (e.g., a Completion Timeout) is detected while in D2, an error

message must be sent when the device is programmed back to the D0 state.

Section 5.3.1.4.1 D3hot State, Page 249 line 27:

5.3.1.4.1. D3hot State

Configuration and Message requests are the only TLPs accepted by a function in the D3hot state.

All other received Requests ~~may optionally~~ must be handled as Unsupported Requests, and all received Completions may optionally be handled as Unexpected Completions. If an error caused by a received TLP (e.g., an Unsupported Request) is detected while in D3hot, and reporting is enabled, the link must be returned to L0 if it is not already in L0 and an error message must be sent. If an error caused by an event other than a received TLP (e.g. a Completion Timeout) while in D3hot, an error message may optionally be sent when the device is programmed back to the D0 state.

Section 5.2, Page 247 line 19, Table 5-1,Note 7:

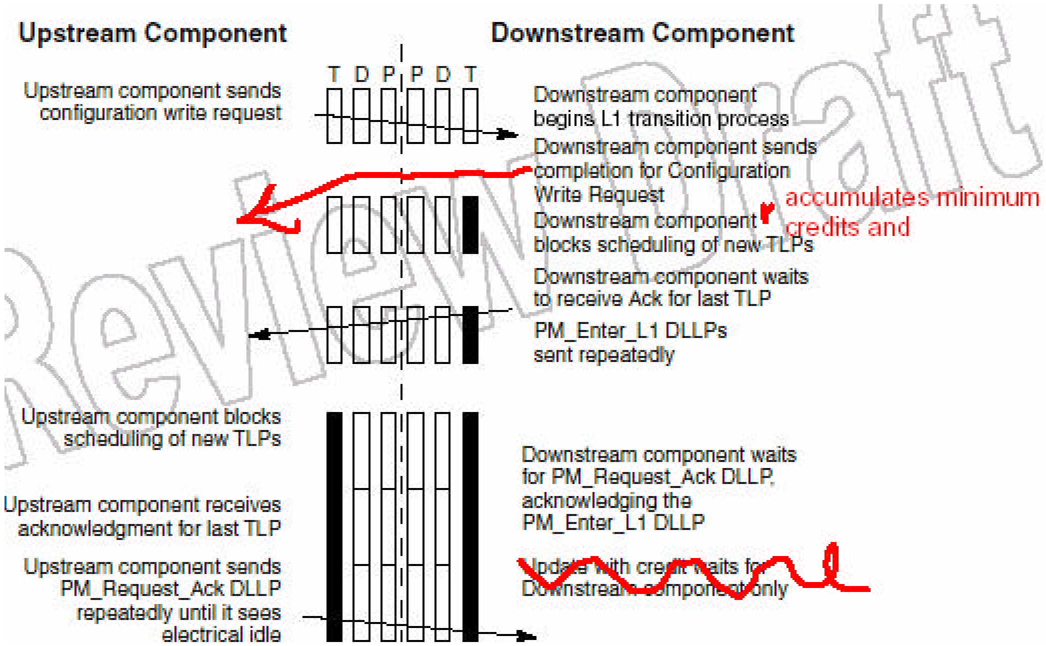
7. Low power mobile or handheld platforms may aggressively reduce power by clock gating the reference clock(s) via the "clock request" (CLKREQ#) mechanism. As a result, components targeting these platforms should be tolerant of the additional delays required to re-energize the reference clock during the low power state exit.

Section 7.8.6, Page 397, bit 18:

Clock Power Management – A value of 1b in this bit indicates that the component tolerates the removal of any reference clock(s) via the "clock request" (CLKREQ#) mechanism when the link is in the L1 and L2/3 Ready link states. A value of 0b indicates that the component does not have this capability and that the reference clock(s) must not be removed in these link states.

This capability is applicable only in form factors that support “clock request” (CLKREQ#) capability.

The text in the figure for the downstream component does not mention waiting to accumulate credits – Edit Figure 5-3 (Sect 5.3.2.1) “Entry Into the L1 Link State” as shown [Note: This figure used to be Figure 5-2; Unchanged parts of figure not shown]:


Upstream Component
Downstream Component
T D P P D T
Upstream component sends configuration write request
Downstream component begins L1 transition process
Downstream component sends completion for Configuration Write Request
Downstream component accumulates minimum credits and blocks scheduling of new TLPs
Downstream component waits to receive Ack for last TLP
PM_Enter_L1 DLLPs sent repeatedly
Upstream component blocks scheduling of new TLPs
Upstream component receives acknowledgment for last TLP
Upstream component sends PM_Request_Ack DLLP repeatedly until it sees electrical idle
Downstream component waits for PM_Request_Ack DLLP, acknowledging the PM_Enter_L1 DLLP
Update with credit waits for Downstream component only

C57. (Cx2) Errata on the 1.1RD – Surprise Down Error

PWG Approved 28 October 2004 -- Release Date: 28 Oct 2004

In Section 3.2.1:

- Upstream components are optionally permitted to treat this transition from DL_Active to DL_Inactive as a Surprise Down error, ~~with the following exceptions~~except in the following cases where this error detection is blocked:
    - If the Secondary Bus Reset in Bridge Control Register has been set to 1b by software, then the subsequent transition to DL_Inactive must not be considered an error.
    - If the Link Disable bit has been set to 1b by software, then the subsequent transition to DL_Inactive must not be considered an error.
    - If a PME_Turn_Off Message has been sent through this port, then the subsequent transition to DL_Inactive must not be considered an error. Note that the DL_Inactive transition for this condition will not occur until a power off, a -reset or a request to restore the link is sent to the Phy layer. Note also that in the case where the PME_Turn_Off/PME_TO_Ack handshake fails to complete successfully a Surprise Down error may be detected.
    - If the port is associated with a hot-pluggable slot, and the Hot Plug Surprise bit in the Slot Capabilities Register is set to 1b, then any transition to DL_Inactive must not be considered an error.
    - If the port is associated with a hot-pluggable slot (Hot-Plug Capable bit in the Slot Capabilities Register set to 1b), and Power Controller Control bit in Slot Control Register is 1b(Off), then any transition to DL Inactive must not be considered an error.

  ~~If the error is masked by any of the above exceptions, either of the following events must cause error unmasking:~~Error blocking initiated by one or more of the above cases must remain in effect until the port exits DL_Active and subsequently returns to DL_Active with none of the blocking cases in effect at the time of the return to DL_Active.

    - ~~A transition to DL_Inactive~~

  Note that the transition out of DL_Active is simply the expected transition as anticipated per the error detection blocking condition.

    - ~~The successful reception and forwarding to the Receive Transaction Layer (see Section 3.5.3.1) of any TLP~~

  If implemented, this is a reported error associated with the detecting port (see Section 6.2).

--- end of errata ---